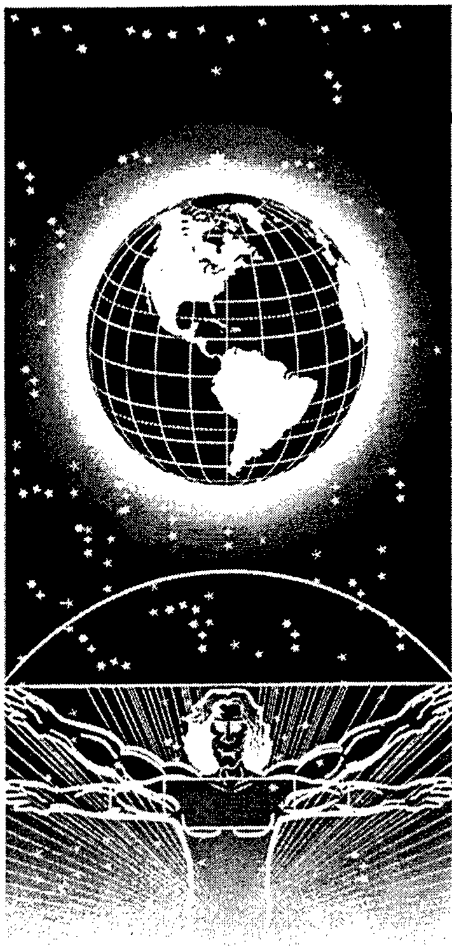
AFRL-HE-WP-TR-2002-0018



UNITED STATES AIR FORCE RESEARCH LABORATORY

The Combat Automation Requirements Testbed (CART) Task 5 Interim Report: Modeling a Strike Fighter Pilot Conducting a Time Critical Target Mission

Bryan E. Brett
Jeffrey A. Doyal
David A. Malek

SCIENCE APPLICATIONS INTERNATIONAL CORPORATION
4031 COLONEL GLENN HIGHWAY
BEAVERCREEK OH 45431

Edward A. Martin
David G. Hoagland
Martin N. Anesgart

HUMAN EFFECTIVENESS DIRECTORATE
CREW SYSTEM INTERFACE DIVISION
WRIGHT-PATTERSON AFB OH 45433-7022

JANUARY 2002

INTERIM REPORT FOR THE PERIOD APRIL 1999 TO OCTOBER 2001

Approved for public release; distribution is unlimited

Human Effectiveness Directorate
Crew System Interface Division
2255 H Street
Wright-Patterson AFB, OH 45433-7022

20020712 030

NOTICES

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2002-0018

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

MARIS M. VIKMANIS
Chief, Crew System Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 2002	Interim, April 1999 - October 2001

4. TITLE AND SUBTITLE
The Combat Automation Requirements Testbed (CART) Task 5 Interim Report: Modeling a Strike Fighter Pilot Conducting a Time Critical Target Mission

5. FUNDING NUMBERS
C: F41624-98-C-6012
PE: 63231F
PR: 2830
TA: 29
WU: 10

6. AUTHOR(S)
* Bryan E. Brett, * Jeffrey A. Doyal, * David A. Malek
** Edward A. Martin, ** David G. Hoagland, **Martin N. Anesgart

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
* Science Applications International Corporation (SAIC)
4031 Colonel Glenn Highway
Beavercreek OH 45431

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
** Air Force Research Laboratory, Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7022

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-HE-WP-TR-2002-0018

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The Combat Automation Requirements Testbed (CART) program is developing and demonstrating human performance modeling technology that can be used to design more effective crew systems. CART applies the task network modeling approach of the Army's Improved Performance Research Integration Tool (IMPRINT) and adds new functions called *goals* that enable more adaptive model performance and a High Level Architecture (HLA) interface for coupling with other simulations. This report documents the first application of CART modeling technology. A model of a strike fighter pilot conducting a time critical target (TCT) mission was developed and integrated with a strike fighter flight simulation that interacted with a Joint Integrated Mission Model (JIMM) TCT scenario. The pilot model was exercised in six TCT scenarios and eight pilots flew the same scenarios in a virtual version of the simulation. Overall, pilots found and destroyed the TCT 98% of the time. The model was successful 100% of the time. Comparison of pilots and the model on summary measures of key mission functions showed generally high correlation between pilot and model performance (Congruency Ratio = .78). The results suggest that human performance models can predict operator performance and provide important insight into operator issues in new systems.

14. SUBJECT TERMS
human performance modeling, task network modeling, goal-oriented performance, pilot models, modeling and simulation, flight simulation, crew system design, mission decomposition, time critical targets

15. NUMBER OF PAGES
234

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. Z39-18

This page left blank intentionally

PREFACE

This effort, "Task 5: *Conduct Two Case Studies*," was conducted under contract number F41624-98-C-6012 with the Crew Systems Development Branch, Crew System Interface Division, Human Effectiveness Directorate of the Air Force Research Laboratory (AFRL/HECI), Wright-Patterson Air Force Base, Ohio 45433-7022, for the period April 1999 to October 2001. Science Applications International Corporation (SAIC), 4031 Col Glenn Highway, Beavercreek, Ohio 45431-7753 was the contractor. Mr. David Hoagland (AFRL/HECI) was the Program Manager. This effort supported Work Unit 28302910, "Combat Automation Requirements Testbed (CART)."

This page left blank intentionally

TABLE OF CONTENTS

This page left blank intentionally

LIST OF FIGURES

This page left blank intentionally

LIST OF TABLES

1.0 INTRODUCTION

This report describes the first case study performed for the Combat Automation Requirements Testbed (CART) program. CART is an Air Force Research Laboratory Human Effectiveness Directorate program that is developing tools that strongly support simulation-based acquisition (SBA) concepts. As an Advanced Technology Development research effort, the goal of CART is to demonstrate and evaluate the application of human performance modeling to the design and development of crew systems that integrate the warfighter and weapon system more effectively.

The demonstrations performed under the CART program occur in the context of case studies. In a case study, a particular human performance context is selected, a model of the human performing in that context is developed, and the human performance model is integrated with a constructive representation of the system, the operator controls, and the mission environment in which the system operates. The integrated constructive testbed is exercised in several scenarios or test conditions and performance data are obtained. In addition to constructive testing, virtual, human-in-the-loop (HITL) simulations are also conducted for the same scenarios and test conditions. Finally, the data resulting from the constructive and virtual testing are compared to determine the extent to which the human performance model predicts actual human performance.

To date, the first of two CART case studies has been completed (Case Study 1). This report provides a brief overview of the CART program and then describes the development, conduct, and results of Case Study 1.

1.1 The Problem CART is Addressing

As the analytical capabilities and potential for cost savings afforded by modeling and simulation (M&S) technology continue to expand, the width and breadth of M&S applications also continue to grow. For many applications, including training, analysis, and acquisition, constructive simulations of systems in their intended environments are proving extremely valuable. Historically, however, one limitation of such constructive simulation environments is their ability to represent the *human* component of the manned system being simulated. While we are generally quite good at representing performance of the hardware and software in a given

system, we tend to struggle in our modeling of the perceptual, cognitive, and physical capabilities of the operator controlling the system. As a result, attributes such as operator workload, performance, and tactics -- each of which are critical components of overall system performance -- are often ignored or severely constrained within the constructive simulation environment, thereby restricting the validity and generalizability of the simulation effort's results. The Department of Defense (DOD), which identified "providing authoritative representations of human behavior" as one of six key goals to be achieved within its modeling and simulation efforts, has recognized this limitation (DOD 5000.59-P).

One particular application of modeling and simulation in which the human representation is often lacking is the area of acquisition. Currently, analysts and decision-makers rely heavily on constructive simulations of a system in its intended environment to help translate mission requirements identified by the warfighter into system performance requirements. Within constructive simulations, sensitivity analyses are conducted on key subsystem attributes by selectively varying attribute levels and measuring the results on mission performance. In this way, performance levels are identified for key subsystem attributes that yield desired levels of mission performance, thereby providing the basis for statements of system requirements.

Unfortunately, consideration of the crew interface as part of the system is generally avoided in these requirements-generation efforts. The acquisition community continues to use expensive HITL simulation, in part because of its inability to model crewmember behavior and human-computer interactions within constructive simulations. Hence, crew interface requirements are not quantifiably linked to the set of overarching measures of weapon system effectiveness as the other subsystem attribute requirements are. Not only can the lack of realistic consideration of the operator on system-level performance requirements lead to crew interface inadequacies, it can also drive *performance and cost* unnecessarily because the simulated system did not represent appropriate tactics. One military analyst recently noted, "Every single analysis that I have ever seen has suffered from the lack of capturing smart tactics. Mistakes such as pursuing an attack when the tactic should have been 'run away' lead to mission outcomes (aircraft loss) that seem to indicate system deficiencies when in fact the system was misused tactically." (Martin, Brett & Hoagland, 1999) Analysts and decision-makers need a means to readily model *and understand*

the effects of human performance on *total* weapon system effectiveness when translating operational requirements into system requirements, and they need to be able to visualize these effects at different levels of aggregation (Martin, et al., 1999).

To address the problem outlined above, the Air Force Research Laboratory initiated the CART program. The program's overall objective is to provide a tool that permits users of constructive simulations to readily develop and integrate human performance models in an effort to achieve more accurate representations of the human operator's impact on overall mission outcomes. More specific objectives of the CART program are to:

(1) advance the state-of-the-art in human modeling using interoperable simulations and practices, such as High-Level Architecture (HLA), (Defense Modeling & Simulation Office, 1998),

(2) demonstrate a robust human modeling architecture that is compatible with current and future DOD simulations,

(3) link operator performance with mission effectiveness, and

(4) provide the capability to trace cause-and-effect relationships during or after simulation runs.

1.2 CART Program Tools

The CART program will extend current constructive M&S testbed capabilities by providing two new tools for enhancing human performance representations in constructive simulation. One is a human performance modeling capability. With this tool, analysts will be able to create models that simulate activities operators would perform in a system. Analysts also will be able to assign parameters to the models to reflect different levels of operator capability. These human performance models will be integrated with constructive models of a system and will interact with the system in the context of a simulated mission. The second tool will provide performance assessment capabilities, supporting generation of measures of operator performance that will be clearly linked to measures of system performance and mission effectiveness. With this tool relationships among operator, system and mission performance will be visualized and traced, and levels of operator performance required to produce desired mission outcomes will be identified.

1.3 CART Program's Phase 1 Components

The current phase of the CART program (Phase 1) consists of six tasks that support the program goals and development of the CART tools. The objectives of each task are discussed briefly below.

Task 1: Crew System Requirements Establishment. The objective of this task is to characterize the current acquisition process for DOD acquisition programs, highlighting the nominal extent to which crew interface requirements are analyzed and established. Essentially, the end goal of Task 1 is to identify the 'user' of CART, the environment and processes in use today, and the most appropriate niche for CART in the acquisition environment.

Task 2: Human Modeling Architecture. Task 2 focuses on defining a human modeling architecture to be integrated with a selected military simulation. It involves the development of a human performance model (HPM) as well as an interface to an engagement-level simulation that both characterizes the flow of information to and from the human operator and provides the appropriate human control interactions to the constructive system model.

Task 3: Conduct Trade Studies. This task involves the conduct of two trade studies to select the two most appropriate operational contexts from among various predefined military domains of interest. The objective of these trade studies is to identify active acquisition programs that are suitable for use in the two different case studies conducted under Task 5.

Task 4: Real-Time Operational Mission Simulation. The objective of Task 4 is to modify and prepare a mission simulator to represent the system and environment for the domains of interest selected in Task 3. This includes adding the needed data collection capabilities for computing mission-level measures of effectiveness (MOEs) down through lower-level human performance measures.

Task 5: Conduct Two Case Studies. Task 5 involves the conduct of both constructive and virtual simulation tests for identical mission tasks within the mission contexts selected under Task 3. These tests will allow comparison of constructive and virtual simulation results to demonstrate

the efficacy of the human performance modeling architecture and interface. The conduct of these case studies represents the heart of activity in the CART program.

Task 6: Prepare Testbed Definition. The final task to be performed under Phase 1 of the CART program is to develop a CART performance specification that defines an implementable testbed. The specification will include guidance regarding how to integrate equipment or instrumentation in a simulation, the software architecture description, Run-Time Infrastructure (RTI) implementation, and a reconfigurable hardware architecture methodology.

To date, Task 1, Crew System Requirements Establishment, has been completed, as has the majority of Task 2, Human Modeling Architecture. Results of these activities, documented in Brett, Doyal, Malek, Martin & Hoagland (2000), defined much of the analytic process and modeling architecture applied in Case Study 1. The following sections of this report focus on the conduct of this case study, discussing the selection of the environment, development of the testbed, test methodology, and results of the data collection and analysis effort.

This page left blank intentionally

2.0 THE CASE STUDY 1 ENVIRONMENT

The purpose of the CART case studies is two-fold: (1) demonstrate the technology of the CART human performance modeling environment, and the approach to model development and integration with constructive simulations, and (2) assess the validity of the CART concept by comparing mission performance data from simulation trials that incorporate a CART-developed HPM with data from trials incorporating a HITL simulator operating in the same environment. If the CART approach and tools can be demonstrated to create and integrate HPMs that successfully represent human performance in the case study environments, CART can be offered as an expanded set of tools to the organization/modeler/analyst seeking to improve representations of operator performance in constructive simulation environments.

2.1 Case Study Selection

A trade study process was used to select a topic for Case Study 1. In the trade study process, potential topics were identified from ongoing or planned acquisitions for new or evolving weapons systems. A brief description of the Case Study 1 environment selection process is provided below. For a more detailed description, see Appendix A. The candidate topics were evaluated in terms of six major factors, described below, that affect the utility of a topic to the CART program:

1. Types of human performance to be modeled. The objective was to select a system in which operators had a significant role in system performance and could affect mission performance -- a system with operator behavior that would be challenging to model and that would test the viability of the CART concept.

2. Availability of existing system/environment models. Funding on the CART program is limited. A key objective was to find a program that had an existing simulation environment that included a constructive representation of the system and mission environment of interest. This would permit us to maximize investment in development of a human performance model and integration of that model with the constructive system/environment simulation.

3. Cost/effort to modify/integrate a constructive simulation. Given the funding limitation noted above, it was important to consider the expected cost of (1) developing a human performance model, (2) making modifications to the constructive simulation to accommodate interaction with the human performance

model, and (3) integrating and testing the resulting human performance modeling testbed to be sure an effective demonstration could be completed within case study funding.

4. Availability of and cost/effort to condition HITL simulators. Given the requirement to compare human model performance with actual human performance, it was necessary to identify simulation environments that possess both an HITL data collection capability and a constructive battlespace environment with which a human performance model could be integrated. Also, the cost of any modifications required to HITL simulation had to be within the constraints of the budget.

5. Availability of data required to generate performance measures. The requirement to compare performance of the human model with actual human performance meant that data had to be available in both the constructive and virtual test environments to support calculation of a common set of performance measures.

6. Program maturity/schedule fit with CART. Case studies had to be accomplished within a given timeframe dictated by the CART contract. It was necessary to find a program within which simulation resources needed by the CART program would be in place and available during the time allotted for Case Study 1.

As described in Appendix A, a variety of programs were considered as case study candidates. Each was evaluated on the above factors. In the end, one program and facility clearly emerged as the strongest candidate. This was the Virtual Strike Warfare Environment (VSWE) hosted in the Aeronautical Systems Center's Simulation & Analysis Facility (SIMAF) at Wright-Patterson Air Force Base, Ohio. The VSWE had been developed to support requirements development for the Joint Strike Fighter (JSF) program. It consisted of a complex, mature, high fidelity virtual simulation and data collection environment for studying effectiveness of conceptual JSFs in a variety of air-to-ground attack missions. It was already developed and its assets were available within the timeframe required by CART. Pilot behaviors required by the JSF and exercised in the VSWE were sufficiently varied and complex to pose a significant human performance modeling challenge. Also, it was determined that the architecture of the flight simulator used in the VSWE was suitable for integrating a human performance model. This permitted the creation of a constructive simulation environment that was very similar to the virtual simulation environment. Consequently, differences observed between the performance of the human model and actual humans could not be attributed to significant differences between the virtual and constructive testbeds.

Beyond the availability of simulation facilities, the SIMAF VSWE offered existing test sets consisting of the following:

- Simulation software builds that represented a specific set of JSF capabilities
- Well-defined scenarios that exercised a broad range of pilot behavior in the context of complex missions
- An extensive set of measures of effectiveness and measures of performance that the JSF program had defined to evaluate system effectiveness
- All the data collection and reduction capabilities required to compute the performance measures

By reusing these test sets, the CART program was able to minimize the amount of development required for the constructive and virtual simulations, and devote more effort to the development of a human performance model of significant complexity.

2.2 The VSWE 3B Mission Environment and Scenario

A number of VSWE exercises have been conducted in support of the JSF program. With each successive VSWE, modeling and simulation components have continued to evolve. At the time the CART program first became involved with the VSWE testbed, the SIMAF and JSF program had just recently completed VSWE exercise "3B." Because of its lower classification level, this exercise provided a more accessible environment for the integration and testing of the CART system. As such, the VSWE 3B environment served as the specific baseline simulation environment for CART's Case Study 1.

2.2.1 The VSWE 3B Environment

The VSWE 3B environment consists of an aircraft simulation with a cockpit that allows HITL control of the aircraft and a mission-level constructive model that provides the mission environment (terrain features, threats, targets, etc.) through which the aircraft simulation flies and interacts. The cockpit is called the mission interactive combat station (MICS), and the Joint Integrated Mission Model (JIMM) provides the mission environment. The JIMM is based largely

upon the Synthetic Warfare Environment Generator (SWEG) and includes capabilities native to Suppressor, the Air Force's long standing mission-modeling environment.

The MICS, shown in Figure 1, is a reconfigurable cockpit simulator that served as the cockpit environment for Case Study 1. The MICS was powered by a SGI ONYX 2 dual rack with 14 R10000 CPUs, two Infinite Reality Pipes (64 MB and 128 MB texture memory), 890 MB RAM and 18 GB of data storage. The cockpit consisted of a 29-inch monitor with touch screen overlays and an F-16 Block 50 stick and throttle. The single-channel out-the-window (OTW) imagery was projected on a screen in front of the cockpit with a head-up display (HUD) overlay. Communication capabilities were integrated between the cockpit and test director area. Audio was generated by the Fighter Requirements Evaluation Demonstrator (FRED) software and I/O boards were installed on each ONYX 2. The MICS uses the JIMM as the environment generation system and it was hosted on a three-processor (R10000) Silicon Graphics workstation with 256 MB RAM and 9 GB hard disk.

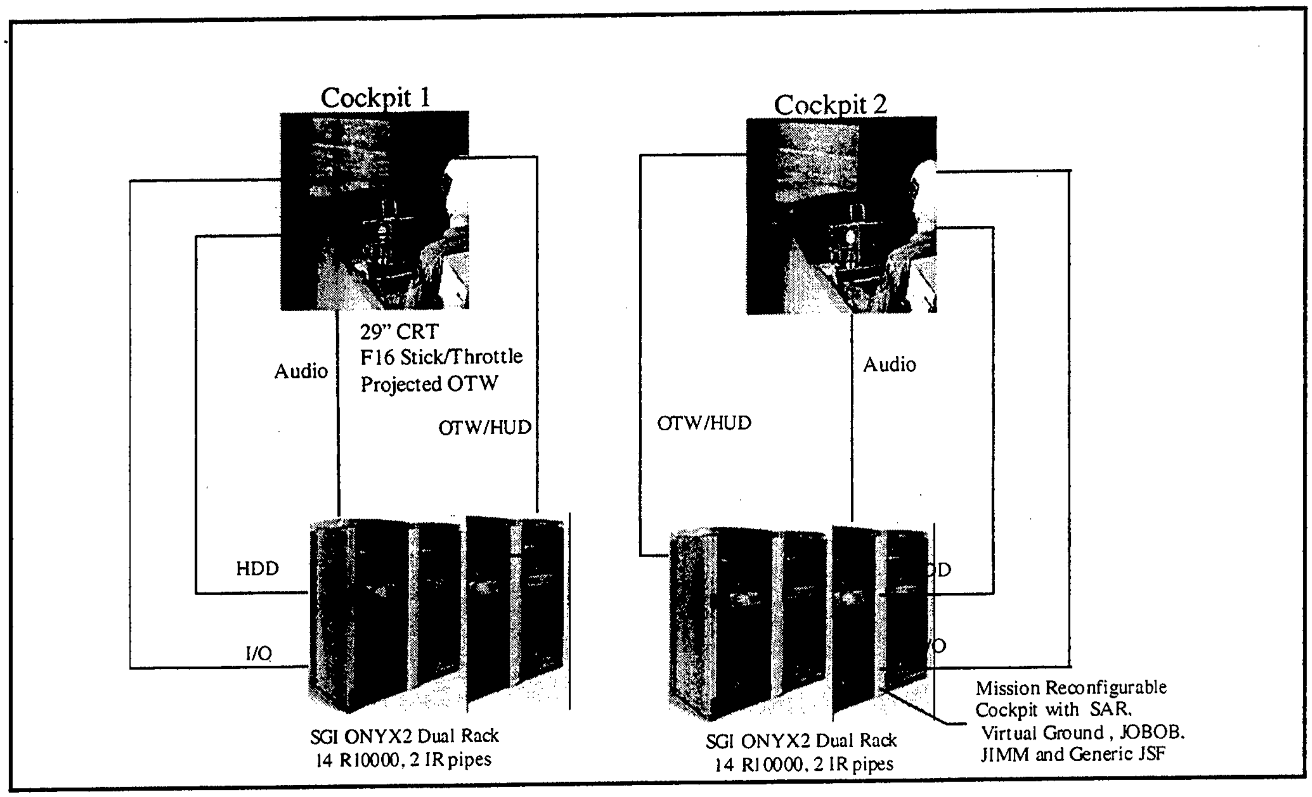


Figure 1. The SIMAF Mission Interactive Combat Stations

During the HITL trials, two cockpits were used to provide data collection from two different pilots flying different missions simultaneously. A closer view of one of those cockpits is shown in Figure 2.

The MICS cockpit is similar to the F-16 in the layout of flight controls and switches, but a major difference between the MICS cockpit and the F-16 is the large head-down display (HDD) presented on the 29-inch monitor.

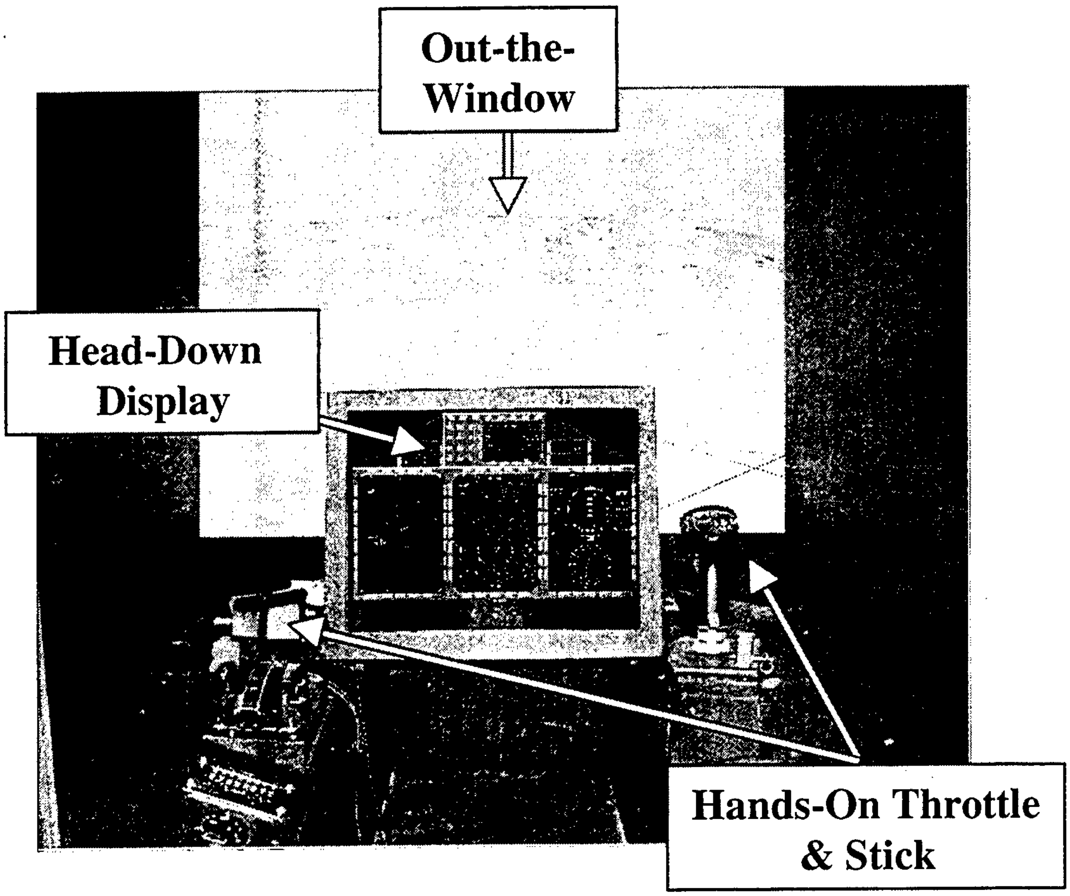


Figure 2. A MICS Cockpit

The primary software for the MICS is comprised of the FRED software, the Camber radar toolkit for Synthetic Aperture Radar (SAR), and Paradigm's Vega for OTW imagery. In addition, an integrated moving map capability, a real-time in-flight route planner, and an effects-level infrared model for targeting were implemented.

The FRED software was used to simulate the generic JSF cockpit environment. The FRED consists of software components that can be used to simulate aircraft systems using various hardware components. For Case Study 1, the FRED software interfaced with a 29-inch monitor and the Block 50 F-16 stick and throttle controls in the MICS.

The FRED consisted of several software modules, each supporting a different area of the simulation environment. The environment models managed the position orientation and velocity of the ownship based on inputs from the flight control system. These models also managed the movement and weapons deployment of all non-ownship entities, and managed the flight of all weapons that were launched by the ownship or by a threat. The sensor models provided data to simulate on-board sensors including an infrared search and track (IRST), electronic support measures (ESM), identification friend or foe (IFF), missile warning radar, and targeting sensors.

The mission computer module served as the core avionics system for the FRED. It provided navigation data and managed various components of the simulation based on pilot input including steerpoint/route management, weapons stores management, tactical sensor management (e.g., sensor modes and fields-of-view), auto modes management, defensive reaction module management, and sensor fusion management. In addition, route management was augmented by a real-time mission planner (RTMP) capability. The RTMP offered the pilot new routes when there were changes in threat situation and/or deviation from the current route exceeded a threshold distance.

The advanced information management system (AIMS) component used the Joint On-Board/Off-Board (JOBOB) sensor fusion software to fuse on-board and off-board sensor tracks. This system filtered out unwanted sensor reports and fusion track files to reduce fusion processing requirements and cockpit display clutter.

The data from the various models within the FRED were displayed on the HDD and the HUD. The FRED simulated the HDD and the up-front control (UFC), which were displayed on the 29-inch monitor. The HDD acted as a multi-purpose display (MPD) and the monitor was equipped with a touch sensitive capability to simulate the functionality of MPD pushbuttons. The HDD display formats included a radar display, a tactical situation display (TSD), a targeting infrared

display, an aircraft management system display (AMS), a moving map display (MMD), and electronic flight instrument (EFI) displays. The up-front controls consisted of touch sensitive areas designed to simulate UFC pushbuttons, a keypad area, and a data entry display. The FRED also provided a simulated HUD that was projected on a screen in front of the pilot station.

The mission environment generator in which the simulated aircraft flew consisted of the JIMM. The JIMM is an event-stepped, object-oriented, general-purpose conflict simulation capable of participating in a network with other simulations, simulators, hardware, and HITL systems. The JIMM was based largely on the Synthetic Warfare Environment Generator (SWEG) and included capabilities native to Suppressor, the Air Force's long standing mission-modeling environment. In Case Study 1, the JIMM model ran the Generic Composite Scenario (GCS) as developed for the JSF VSWE 3B. The GCS represented the specified mission environment including the physical aspects, physical influence, disruption, and movement of objects within the environment (e.g., the target, other moving objects, roads, buildings, terrain features, etc.).

2.2.2 The Time Critical Target Scenario

The VSWE 3B exercise examined JSF performance in several scenario contexts, but of particular interest in CART Case Study 1 was the attack of a time critical target (TCT). TCTs are high-value, fleeting targets such as tactical ballistic missile launchers. The TCT attack mission, as demonstrated in Operation Desert Storm, has proven quite challenging for strike fighters due to its time-constrained nature and the pursuit of a relatively small, mobile target whose location is uncertain. Figure 3 depicts the general form of the TCT scenarios used in VSWE 3B. The scenario calls for the pilot of a strike aircraft to employ multiple sensors to acquire and attack the mobile target. These sensors include real beam and SAR with ground moving target indication (GMTI), as well as a targeting infrared (TIR) system. During ingress, the pilot is required to evade a pop-up threat that launches a surface-to-air missile (SAM) and to subsequently recapture the ingress route and resume target acquisition. In addition, the pilot is required to receive and act upon an in-flight target intelligence update that provides a more accurate representation of the target's position. If the target is successfully acquired prior to arrival at the planned weapon release point, the pilot then attacks it. Otherwise, the pilot is required to perform a manual re-

planning activity in which a new route to refly the target area is developed. Once on this 'refly' route, the pilot continues attempting to acquire and attack the target.

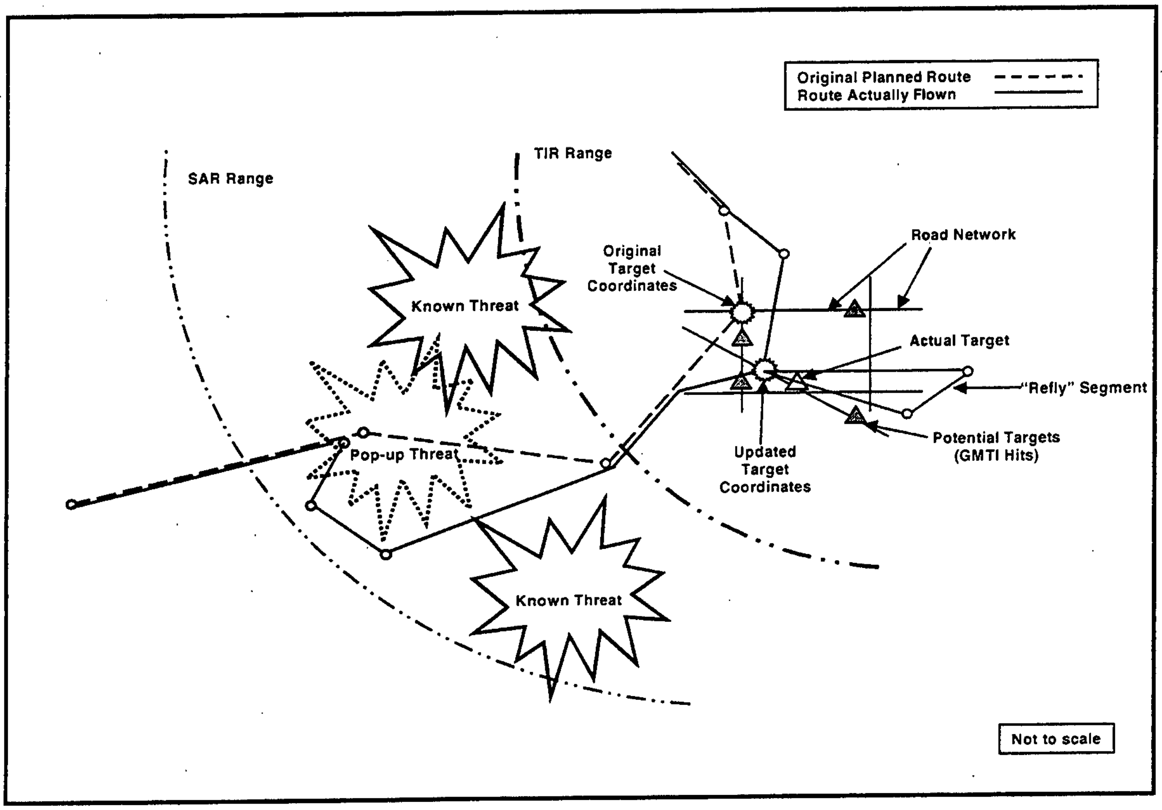


Figure 3. Illustration of Key TCT Mission Components

It is this scenario that drove the high-level requirements for developing the Case Study 1 HPM. The HPM development effort, described in Section 3.0, was focused on creating a model that could realistically represent pilot behavior and decision making associated with performing the strike mission in the above scenario.

3.0 TESTBED DEVELOPMENT

Once the Case Study 1 mission environment and scenario had been fully defined, the CART testbed development effort was initiated. This effort began with a mission decomposition that served to break down and organize the mission into the various operator goals, functions, and tasks performed during the course of the specified mission. Using the CART software, the goals, functions, and tasks were then implemented in a task network model that represented the function/task hierarchy and specified decision rules regarding the sequences of goals, functions, and tasks to be performed. To complete the model, each task was then characterized in terms of task time and accuracy, release conditions and effects, and operator workload. In addition to the HPM development effort, modifications were made to the cockpit/environment simulation to enable the sending and receiving of data and commands to and from the constructive simulation environment. These activities, focusing on testbed development, are described at a high level in the sections below. The actual details of the HPM developed for Case Study 1 are contained in appendices to this report.

3.1 Mission Decomposition

The effort to determine pilot goals and activities associated with the specified scenario began with a detailed mission decomposition. It should be pointed out that the decomposition focused on and represented only those goals, functions, and tasks relevant to a pilot flying in the *VSWE 3B part-mission simulation* environment. There were a number of constraints and simplifications in this environment that would not apply in a real-world flight task. For example, there were no system malfunctions in the particular VSWE scenario, nor was there a possibility for an air-to-air encounter. In addition, the simulated part-task mission was flown at 30,000 feet and did not require a takeoff or landing. As such, many pilot activities associated with these events (e.g., visual scanning of the sky for other aircraft, visual scanning of the ground for targets, checking of engine temperature, communication with air traffic control) were not generally performed by pilots in the simulator. Because results of the HPM trials were to be compared with that of HITL data from the simulator, the decomposition and subsequent HPM

also omitted these types of activities, focusing only on those activities required by the part-task simulation environment.

3.1.1 Means-Ends Decomposition.

The mission decomposition was based loosely upon Jens Rasmussen's "means-ends" hierarchy, and was intended to identify and organize attributes of the mission that were to be subsequently incorporated into the HPM (Rasmussen, Pejtersen, & Goodstein, 1994). The organization of these attributes is hierarchical and includes levels reflecting the mission purpose, operator goals, functions, tasks, and physical systems with which the tasks are performed. To develop the Case Study 1 mission decomposition, modelers began by conducting a series of interviews with a USAF strike fighter pilot. This pilot had experience in the F-15E aircraft, had served as a subject in the VSWE studies in support of the JSF program, and had subsequently served as a subject matter expert (SME) for the VSWE studies. Thus, he had a unique combination of insights regarding the VSWE mission environment, aircraft cockpit, and scenario-specific tactics. During these interviews, the SME described pilot goals, functions, tasks, procedures and decisions relevant to performing the specific VSWE 3B mission. Based on information obtained in these interviews, a baseline mission decomposition was created. It was subsequently presented to the SME and to two USAF F-16 pilots for review and comments. In addition, modelers consulted SMEs familiar with the cockpit environment used in the VSWEs and also reviewed the VSWE 3B Pilot's Manual to identify and understand the specific equipment and procedures used to perform the specified pilot tasks in the simulator.

The mission purpose, goals, and functions identified through the decomposition process are illustrated in Figure 4. The purpose (to destroy the TCT) is supported by five pilot goals. The *Control Aircraft / Maintain Situation Awareness* (SA) goal represents the pilot's goal of monitoring his instruments to maintain awareness of the aircraft and mission status. The *Control Aircraft / Maintain SA* goal can be thought of as the 'default' or *Mission-level* goal that is typically active throughout the mission. The concept of operations for the aircraft calls for autopilot flight in non-evasion situations, and thus, this goal does not include making control inputs to the aircraft. The monitoring functions include listening to the audio channel, checking the flight instruments, monitoring mission progress and threats (i.e., the tactical situation), and

checking the aircraft system status. The *Evade Threats* goal represents the pilot's goal once a missile has been launched at the aircraft. It consists of monitoring the audio channel for additional threat tones, evaluating the severity of the threat, selecting an evasion strategy, executing that strategy through a manual maneuver and use of countermeasures, and returning the aircraft to normal auto-pilot flight upon completion of the evasive action. The *Navigate* goal reflects the pilot's goal of changing the desired mission route. Within the overall scenario, this goal includes functions of accepting a re-plan generated automatically by the mission planner or requesting a manually generated mission re-plan. Manual re-plan functions include configuring the planner, creating a plan to refly the target area if the target is not detected and identified on the first pass, creating a plan to attack the target once it is identified, or planning to return to base (abort) after an attack or after two failed attempts at detection and identification. The *Acquire Target* goal consists of functions that support sensor employment for target acquisition. These include updating target coordinates, choosing a sensor and deciding where to aim it, imaging the target, evaluating the resulting image, updating the shootlist, and designating the object if it is identified as the target. Finally, the *Attack* goal reflects the pilot's desire to maintain sensor track on the target until it is within the weapon release envelope, and to subsequently release a weapon on the target.

Figures 5 through 9 illustrate the lower levels of the mission decomposition for the *Control Aircraft/Maintain SA*, *Evade Threats*, *Navigate*, *Acquire Target*, and *Attack Target* goals, respectively. Each of these figures shows the pilot tasks performed in support of the identified functions as well as the interface (physical form), if applicable, with which the pilot performs the task. The numerous functions and tasks identified in the up-front mission decomposition will not be discussed in detail here; however, each function represented in the final HPM is briefly described and diagrammed Appendix B.

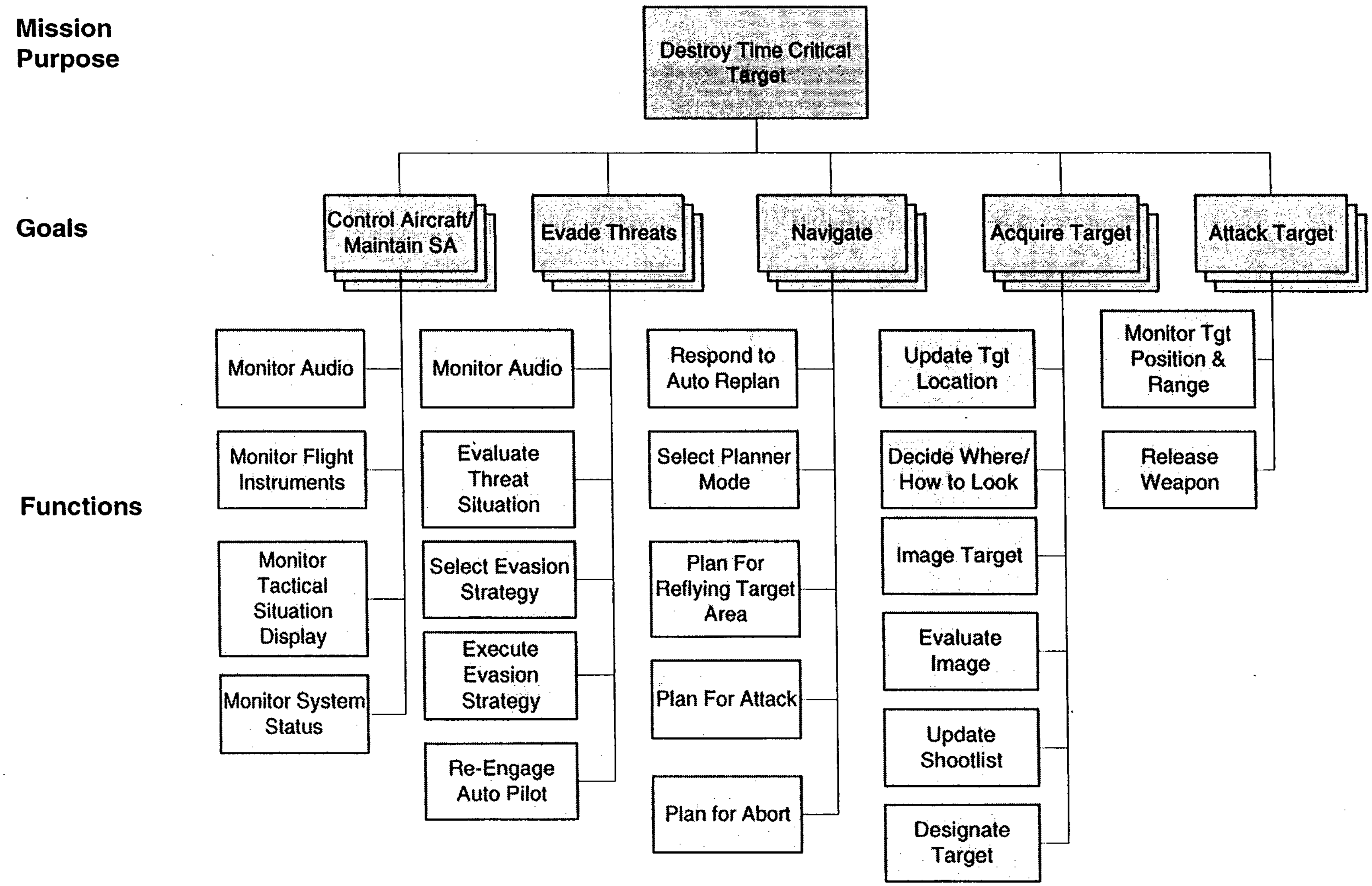


Figure 4. Mission, Goal, and Function Levels of the Mission Decomposition

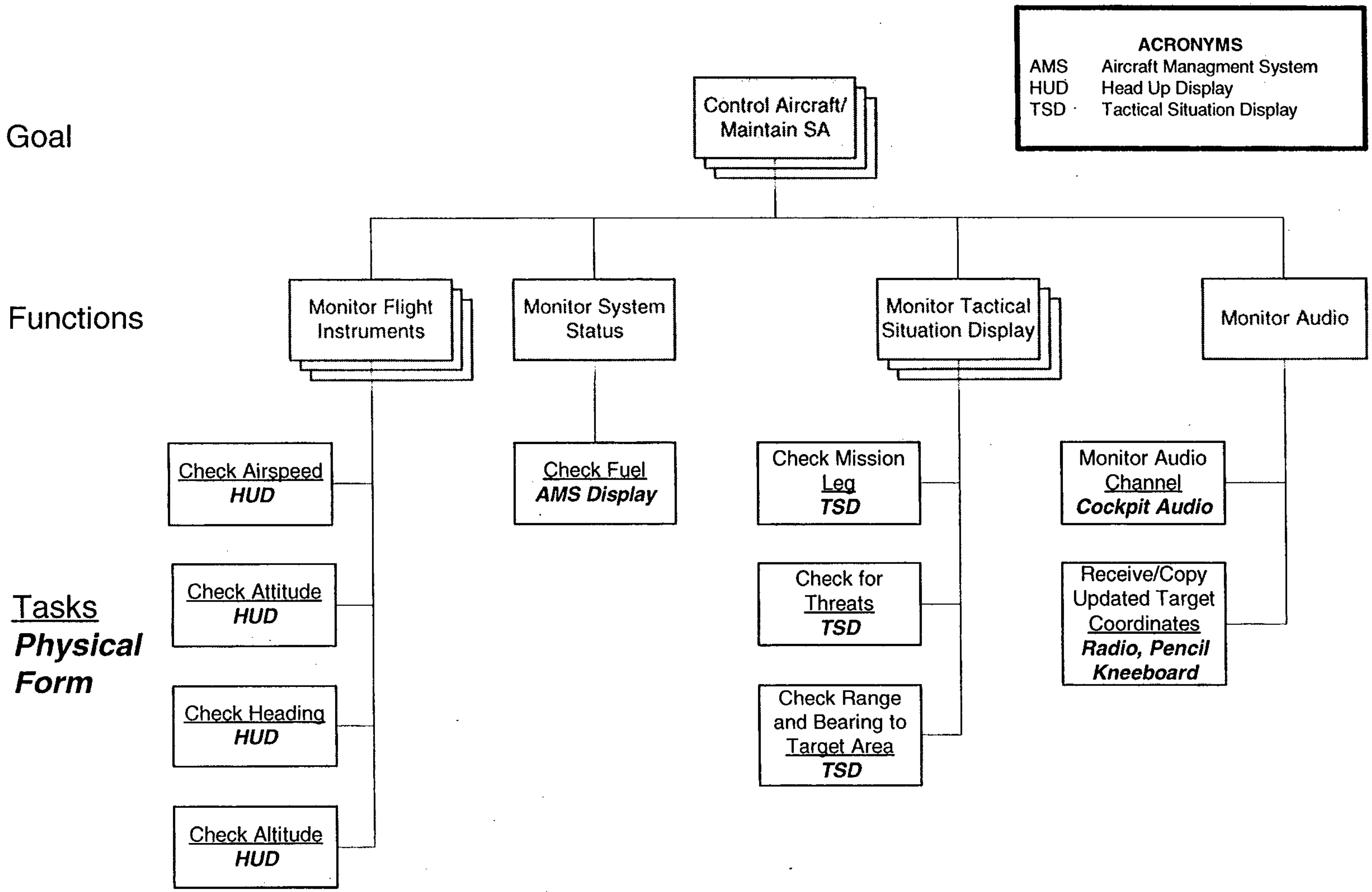


Figure 5. Decomposition of the *Control Aircraft / Maintain SA* Goal

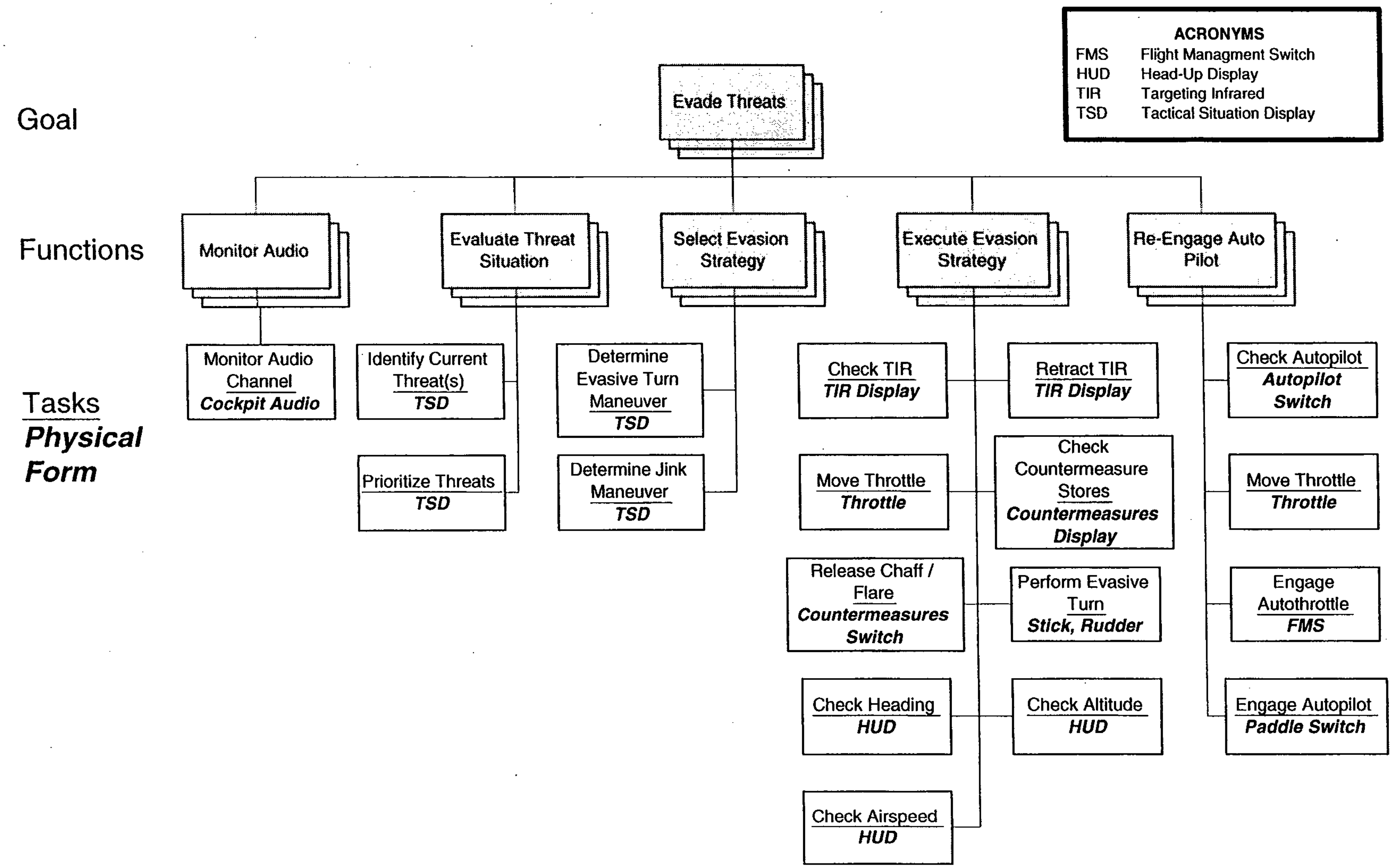


Figure 6. Decomposition of the *Evade Threats* Goal

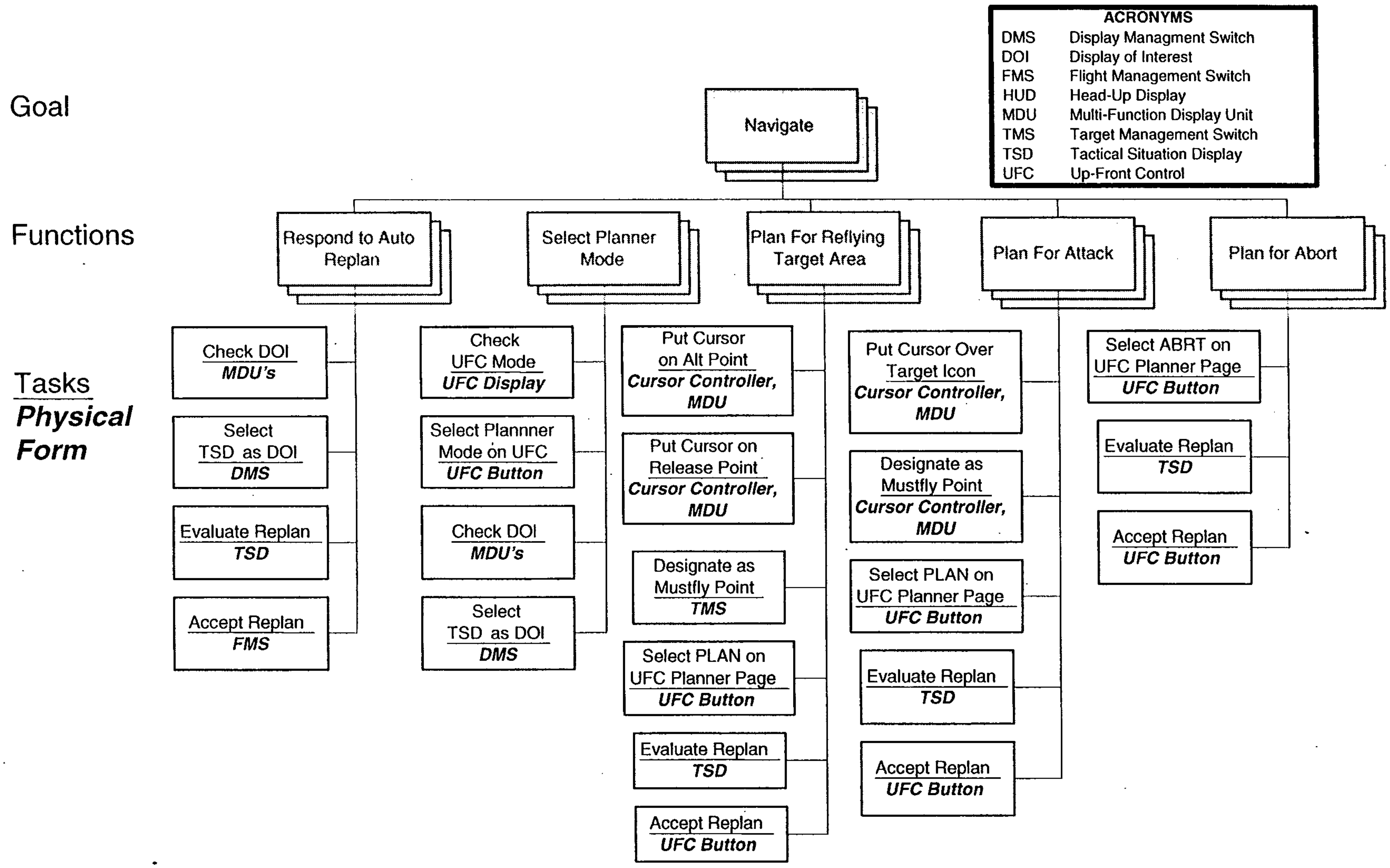


Figure 7. Decomposition of the *Navigate* Goal

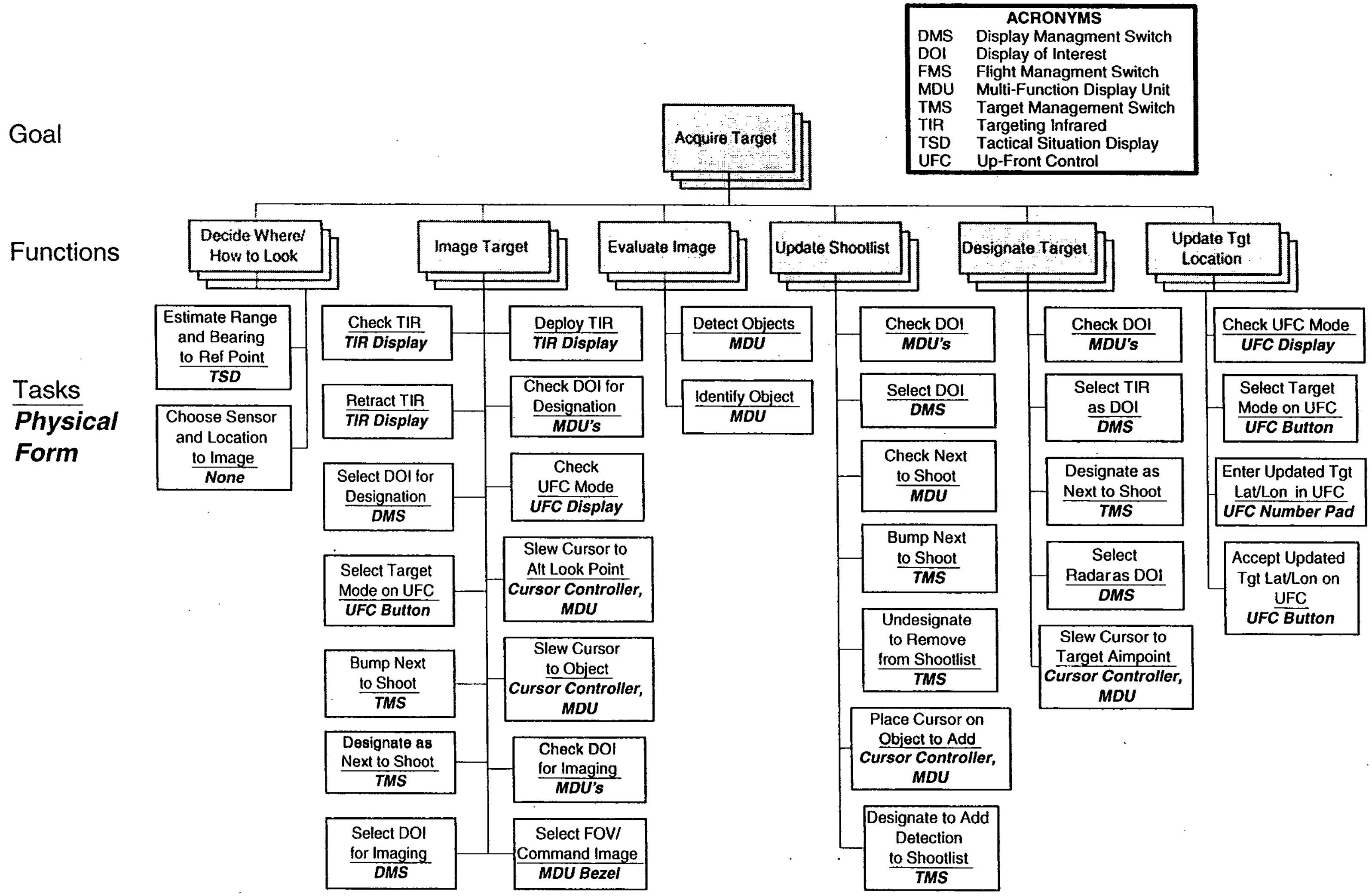


Figure 8. Decomposition of the *Acquire Target* Goal

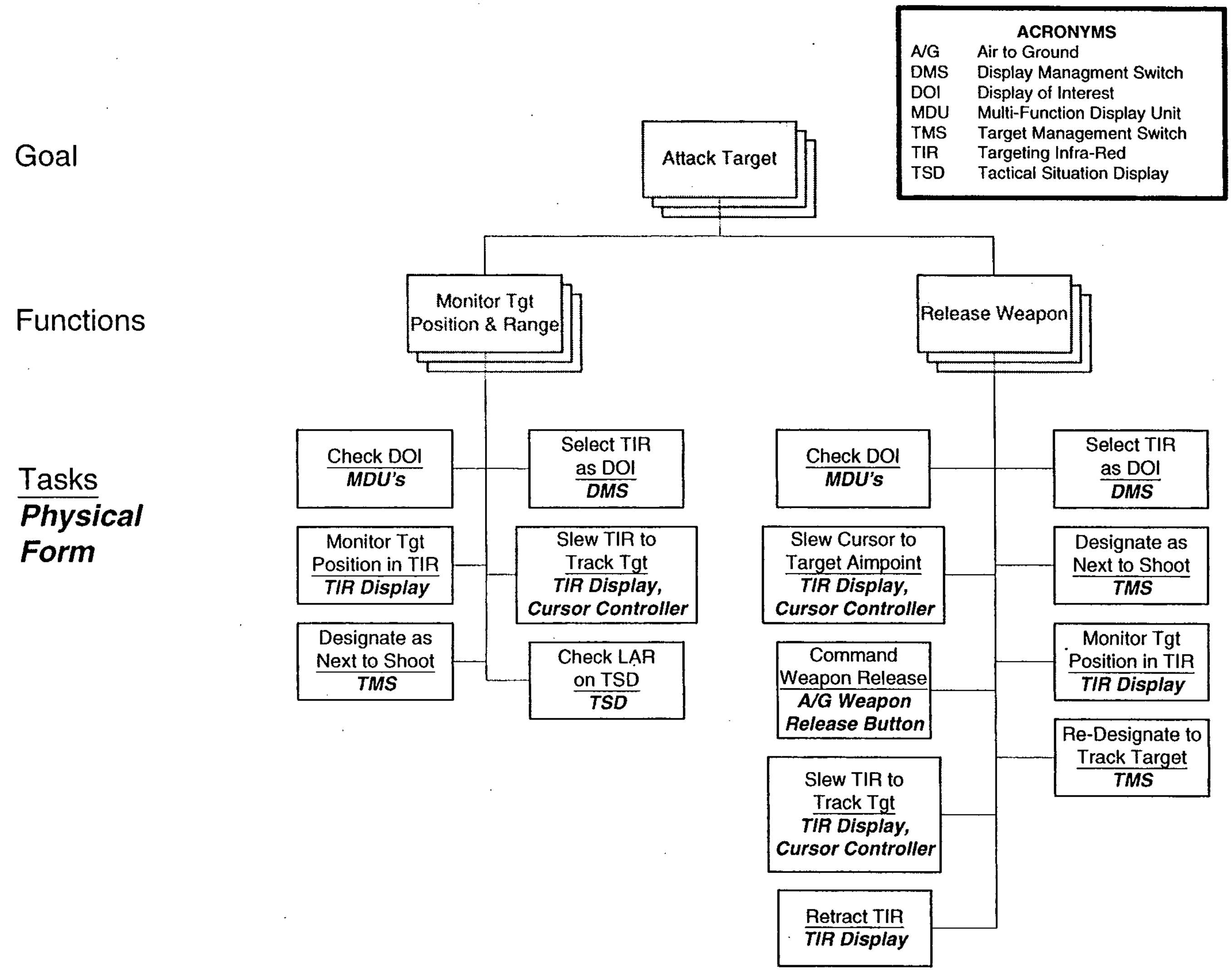


Figure 9. Decomposition of the *Attack Target* Goal

3.1.2 Identifying Task Inputs, Decision Logic, and Pilot Commands

Once pilot goals, functions and tasks had been identified, the process of identifying task information inputs, decision logic, and command outputs began. Subject matter expert interviews and the VSWE 3B Pilot's Manual were used to identify what information is required by the pilot to perform the various tasks, how that information is used to make decisions, and the results of those decisions in terms of pilot interactions with the aircraft (i.e., pilot commands). This information would subsequently serve as input for developing task network model diagrams, for populating task effects, and for defining decision criteria in the HPM. For each task that was identified as having an associated pilot decision or command, the pilot information requirements, decision process and/or resulting pilot action were identified. An example of this information is shown in Table 1. A full listing of the initial information, decision logic, and command requirements identified for inclusion in the model is presented in Appendix C.

Table 1. Example of the Information, Decision, and Pilot Commands Identified for a Given Task

Goal	Function Name	Task Name	Information In	Decision	Command Out
Evade	Execute Evasion Strategy	Maintain Evade Maneuver	Current heading, desired evasive heading	If current heading = desired evasive heading, then end the turn. Else, continue to turn.	Continue Evasive Turn

3.2 Human Performance Model Development

The completed mission decomposition served as the basis for developing the human performance model. The operator goals, functions, tasks and equipment (physical forms) identified in the decomposition, as well as the rules for how they interact and how information is used over the course of a mission, defined the primary attributes that the model needed to possess. The sections below discuss the steps involved in implementing the knowledge gained in this decomposition process to develop the HPM used in Case Study 1.

3.2.1 Diagramming the Network.

As discussed earlier, the CART concept adopts a task network approach to modeling and makes use of a specially tailored task-network modeling environment (i.e., the CART software) to develop human performance models. At the goal level, the HPM represents the operator objectives and priorities that organize behavior. At the lower, function and task levels, model nodes and sequences represent the operator's procedural knowledge. The model development process began with diagramming the goals, functions, and tasks identified in the decomposition to form a series of function/task networks. Each goal, function, and task was represented with a single goal node, function node, or task node, respectively, in a network diagram. These nodes were structured hierarchically, with goal nodes decomposed into function nodes, and function nodes decomposed into task nodes. The nodes were then connected with arrows to specify the sequence of activities that might occur within a given function. In many cases, multiple branches or pathways emerged from a node, reflecting potential branches of multiple, probabilistic, or tactical decisions. Tactical decision pathways represented the pilot's decision options, allowing alternate paths through the function/task networks based on the 'state of the world' and the pilot's goals at the time a task node was executed. An example of a task network diagram can be seen in Figure 10. An explanation of the task network symbology and the full set of task network diagrams comprising the Case Study 1 HPM are contained in Appendix B.

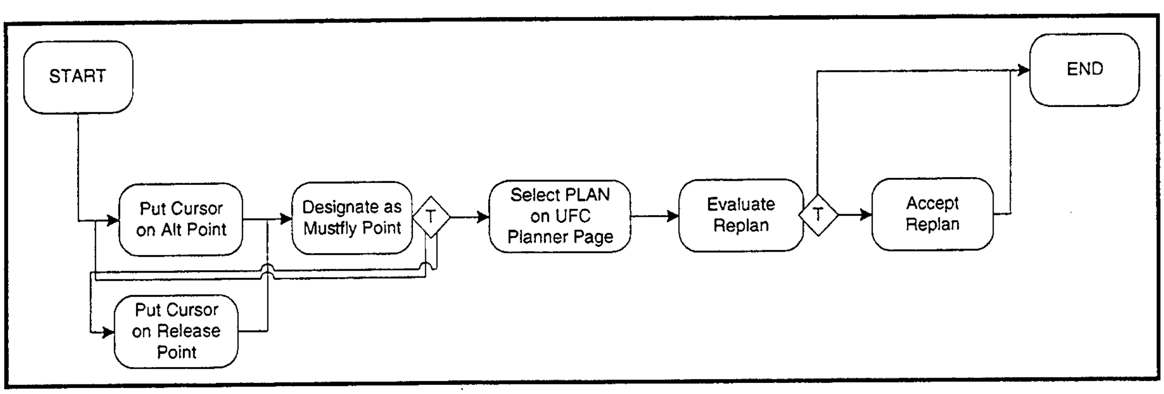


Figure 10. Sample Task Network Diagram

As with the task decomposition, the sources of information for linking network nodes and inserting tactical decisions were the pilot interviews and the VSWE 3B Pilot's Manual.

During this step, it became necessary to add a number of 'dummy' task nodes to the model. These dummy nodes, which were subsequently assigned zero time and workload, were not considered operator tasks, but rather were inserted to allow more modeling flexibility such as embedding tactical decisions among functions and the rejoining of multiple pathways.

3.2.2 Task Performance and Workload

The next step in developing the HPM was to populate each of the task nodes in the network with data that characterized the operator performance for that task. The modeling environment allows this characterization across a number of dimensions including a time standard, accuracy standard, accuracy measure, mean time, standard deviation time, distribution type, mean accuracy, standard deviation accuracy, and workload values encompassing the visual, auditory, cognitive and psychomotor dimensions. Because the model was to be integrated with a constructive simulation environment that, in effect, provided the standard or boundary conditions, defining time/accuracy standards was not necessary. For the development of this model, efforts focused on characterizing only the mean task time and the multidimensional workload values for each task. (Task accuracy was set to 100%.) A listing of all non-dummy tasks and their associated time and workload values is presented in Appendix D.

Values for task times were primarily derived using the micromodels resident within the CART software. Based on the level of detail at which the task was modeled, the task time reflected either the time associated with a *discrete* task represented in a micromodel (e.g., a button press) or a *combination* of tasks (e.g., eye movement + fixation). In cases where an appropriate micromodel did not exist in the software, task times were assigned using actual data from observation of pilots performing the task in the VSWE cockpit. Further, a small number of task times were entered as expressions. These values changed as a function of some variable in the model. For example, the time assigned to the threat prioritization task varied as a function of how many threats were currently active.

Next, workload values were assigned to each of the tasks in the model. For each of the workload dimensions, the CART software provides a set of seven workload values ranging from 0.0 to 7.0. Associated with each workload value is a verbal description of the type of

human activity that corresponds to the given value (e.g., **Psychomotor**: 2.20 -- Discrete Actuation (button, toggle, trigger)). For each task in the model, a verbal description of the type of visual, auditory, cognitive, and psychomotor activity that most closely describes the task was identified. Its corresponding workload value was then assigned to the task.

3.2.3 Coding the Model

The next step in the model development process involved coding the model. Model coding, described below, consisted of a number of programming activities that defined the internal processes of the model.

3.2.3.1 Variable Definition and Mapping. Within the model, two types of variables are defined. *External* variables are those that are used by both the HPM and the constructive simulation environment. In this model, they include such things as entity positions and states as well as commands that are sent from the HPM to the constructive environment (e.g., aircraft position data and cockpit control inputs). *Internal* variables are those used only within the HPM (e.g., 'perceived' airspeed, highest priority threat). Using a 'mapper' capability in the CART software, variable names for external variables defined in the model were then mapped to their corresponding variable name in the constructive simulation. Appendix E provides a list of all HPM variables.

3.2.3.2 Specifying Release Conditions, Effects, and Decision Rules. For each task, any release conditions, beginning and/or ending effects, and decision rules were specified. Release conditions are expressions that must evaluate to 'true' before the task can fire. For example, a release condition in the *Evaluate Re-plan* task is that a new re-plan must be available for viewing. Beginning and ending effects are expressions that execute at a task's onset or conclusion, respectively. In the HPM, these effects are often used to set the value of a 'perceived' variable equal to that of its truth data counterpart, to generate a 'command' to be sent from the HPM to the constructive simulation, or to call a macro containing more complex procedures.

In addition, for any task from which two or more pathways emerged, decision node logic had to be specified. Decision nodes use probability assignments in which the probability of

executing each potential pathway is assigned, or tactical decisions in which expressions using internal or external variables are evaluated to select a particular pathway. A complete listing of the actual code associated with model release conditions, the effects (excluding macros), and the decision nodes is presented in Appendix F.

3.2.3.3 Defining Macros. Often, when a relatively complex set of code was required to perform a set of calculations, this code was entered into a user-defined macro. In the HPM, such macros were used for initializing variables, specifying look point data for the acquisition process, and calculating the detection and identification of objects in the sensor field of view. A brief description of all the macros used in Case Study 1, as well as their actual code, is contained in Appendix G.

3.2.3.4 Defining Priorities, Action Rules and Goal Triggers. Another coding activity involved specifying how the various goal functions interact and how they get triggered. Goal function interaction is managed, in part, through a goal action matrix in the software. This matrix is used to prioritize the goal functions and to specify the effect of a newly triggered goal function on the Mission-level network and on any other goal function networks that are currently running. Newly triggered goal functions can be assigned to run simultaneously with other goal functions, to suspend any subordinate goal function(s), or to abort any subordinate goal function(s). Table 2 shows the goal action matrix developed for the Case Study 1 HPM.

Table 2. Goal Action Matrix for Case Study 1 HPM

Newly Active Goal	Action Upon Competing Goals				
	Mission	**Evade**	**Navigate**	**Acquire**	**Attack**
Evade	Interrupt	Nothing	Abort	Abort	Abort
Navigate	Nothing	Nothing	Nothing	Interrupt	Interrupt
Acquire	Nothing	Nothing	Nothing	Nothing	Interrupt
Attack	Nothing	Nothing	Nothing	Nothing	Nothing

The first column in the matrix shows the prioritized goal functions, listed in order from highest to lowest priority. The SMEs deemed *Evade* the highest priority goal function. Given the scenario, they felt that pilot/aircraft preservation was more important than target prosecution such that all other ongoing activities should get interrupted or aborted once a missile is in the air and evasive action is required. The *Navigate* goal function was assigned the next highest priority as it had a direct impact on both threat avoidance and target acquisition. When the *Navigate* goal function is triggered, the *Acquire* and *Attack* goal functions get interrupted. The *Acquire* goal was given the next highest priority since a successful attack is dependent upon target acquisition. For the purposes of the matrix, a newly triggered *Acquire* goal is assigned to interrupt target *Attack*. In practice, however, the *Attack* function always follows the *Acquire* goal such that they are never active at the same time.

Another goal management activity involved specifying the trigger conditions for each goal. Trigger conditions consist of expressions that, when evaluating to 'true', trigger the onset of a goal function. In the model, they often include statements that evaluate the physical state of the world (e.g., range to the target area <= 20NM) and the status of the mission (target found = FALSE). In addition, however, trigger conditions must also evaluate current status of other higher priority goal functions (*Evade* goal function = not running). In the version of CART software used for Case Study 1 model development (CART version 1.05G), the goal action matrix only specified the impact of a *triggered* goal function on other, currently running goal functions[1]. It did not consider *whether to trigger* a goal function based on the current status of these other goals. Thus, goal function trigger conditions used in the Case Study 1 HPM had to also include arguments to evaluate the status of higher priority goal functions, and to trigger the new function only if its action was compatible with (i.e., would not be suspended or interrupted by) the higher priority function.

[1]The CART software goal management scheme has since been modified to evaluate the status and action rules of a higher priority goal function prior to starting a lower priority function. This change first appeared in CART Software Version 1.07F.

3.2.3.5 Employing a Workload Management Scheme. The last step in the model coding process involved implementing a means of managing pilot workload during the course of the mission. The goal management structure allows two goals to be active simultaneously. That is, multiple functions and tasks that support these goals can fire at the same time. Humans, however, are often limited in their ability to perform multiple tasks simultaneously. For example, they cannot examine two separate visual display screens at the same time. To better reflect this limitation in the model, a workload management scheme was implemented in goal functions that had the potential to run simultaneously. This was implemented as a task management scheme that represented the deliberate distribution of visual resources across tasks residing in multiple goal functions.

Based on the goal management matrix described above, the Mission-level model can run simultaneously with the *Navigate*, *Acquire*, or *Attack* goal functions. Thus, the Mission-level model needed a means of time-sharing with these goal functions. To accomplish this, dummy tasks, internal variables, and release conditions were used to force the mission network and any competing goal function network to 'take turns' when executing task loops. For example, consider the case in which the Mission-level network and the *Acquire* goal function network are running simultaneously. At the onset of a Mission-level network loop, an internal Mission workload variable is set to 'true'. When the loop concludes, this variable is set back to "false." Meanwhile, the release condition for the first task in the *Acquire* goal function is that the Mission workload variable be set to 'false'. Thus, the *Acquire* goal function can only begin after the current scan of the instruments is completed in the Mission-level network. Once the Mission-level loop is completed and the *Acquire* loop is initiated, an *Acquire* workload variable is set to 'true'. The *Acquire* workload variable remains 'true' until the *Acquire* loop is completed (or until the 'pilot' requests a SAR image, for which he must wait a number of seconds), at which time it is set back to 'false'. The initial task in the Mission-level network loop also has as a release condition that the goal function workload variables, including the *Acquire* workload variable, must be set to 'false'. Therefore, as the *Acquire* function completes a loop through the network and its workload variable is set to 'false', the Mission-level network is once again free to begin another loop. This time-sharing process continues as long as the *Acquire* function is active.

3.3 HLA Interface

Once the HPM was developed, it was integrated with the constructive simulation environment (FRED/JIMM). Communication between the HPM and the FRED/JIMM simulation occurred via the HLA RTI. For this effort, the Defense Modeling and Simulation Office (DMSO) HLA RTI version 1.3 was used. The HPM received data regarding system and mission status from the constructive system simulation. Actions to be implemented by the system (e.g., maneuver, target designation, weapon launch) were passed to the constructive simulation by the task network model.

CART's HLA interface employs the Real-time Platform Reference Federation Object Model or RPR FOM (Simulation Interoperability Standards Organization, 1999). While CART modelers are able to readily access entity state data directly available in the RPR FOM, CART makes extensive use of the FOM's Simulation Management (SIMAN) Interactions capability. SIMAN Interactions are used to pass HPM-unique data (e.g., information displayed on operator interfaces and inputs to operator controls) between the HPM and the system it is controlling. A graphical user interface enables a user to define SIMAN Interaction data packets for a given HPM during model development. The CART RTI middleware uses these SIMAN definitions to conduct the data exchange with the constructive simulation. Thus, CART RTI middleware can remain constant while the HPM changes or as the CART tool is used to develop new models and integrate with new constructive simulations.

The CART team chose to utilize the RPR FOM over creating new FOMs or adopting an Agile FOM Framework (AFF) for the following reasons:

(1) using the SIMAN Interactions, the RPR FOM can quickly and efficiently be adapted to send much of the non-standard data that will quite often be communicated in a CART Federation,

(2) with the RPR FOM, CART users do not have to possess the programming skills required to develop and maintain middleware code,

(3) the AFF only works if a Federate's simulation object model (SOM) is similar to the FOM of the Federation (i.e., for the mapping to work, the SOM must have a corresponding concept within the FOM),

(4) the RPR FOM is well thought out, tested, and reliable,

(5) the RPR FOM is currently available with no development or maintenance expenses, and

(6) the RPR FOM will most likely be the FOM of choice for many Federates with which a CART user will want to interact.

3.4 Constructive Testbed Modification/Development

Model integration with the constructive simulation also required modifications to the FRED/JIMM component (see Figure 11). As described in Section 2.2, the FRED provides the virtual cockpit management software, controls the avionics and aero models, and manages the controls and displays. The JIMM is an event-stepped, mission-level modeling environment that provides the terrain, targets, and threats, as well as all command, control, computer, communications, intelligence, surveillance, and reconnaissance (C4ISR) data and events. The shared memory interface is an area that allows assets external to JIMM to dynamically interact with JIMM. This area typically resides in a Shared Common Random Access Memory Network (SCRAMNET) shared memory. In order to interface the VSWE simulation with the HLA RTI and the HPM, a number of modifications and extensions to the VSWE testbed were required.

The primary software changes to the VSWE environment resided in the FRED component. These changes included the addition of a new task and minor modifications to the executive and mission computer. The most significant modification to the FRED was the addition of a new scheduled task to the FRED executive configuration file. The major focus of the new CART task was to add a layer of software that would supply simulation data to, and interpret commands from, the HPM. The subsystem, comprised of the FRED/HPM interface software and the HLA RTI interface software, performs two major functions; these are data exchange and time management. Via calls to the HLA RTI interface, the FRED/HPM interface software retrieves and packages simulation data to be sent to the HPM, receives and

processes commands from the HPM, and interfaces with the FRED simulation utilizing several new packages that allow the HPM to interact with the aircraft (i.e., actuate controls in the cockpit and perceive information available on cockpit displays). The HLA RTI interface software provides the interface to the HLA RTI, sends and receives the actual data packets, and facilitates time management.

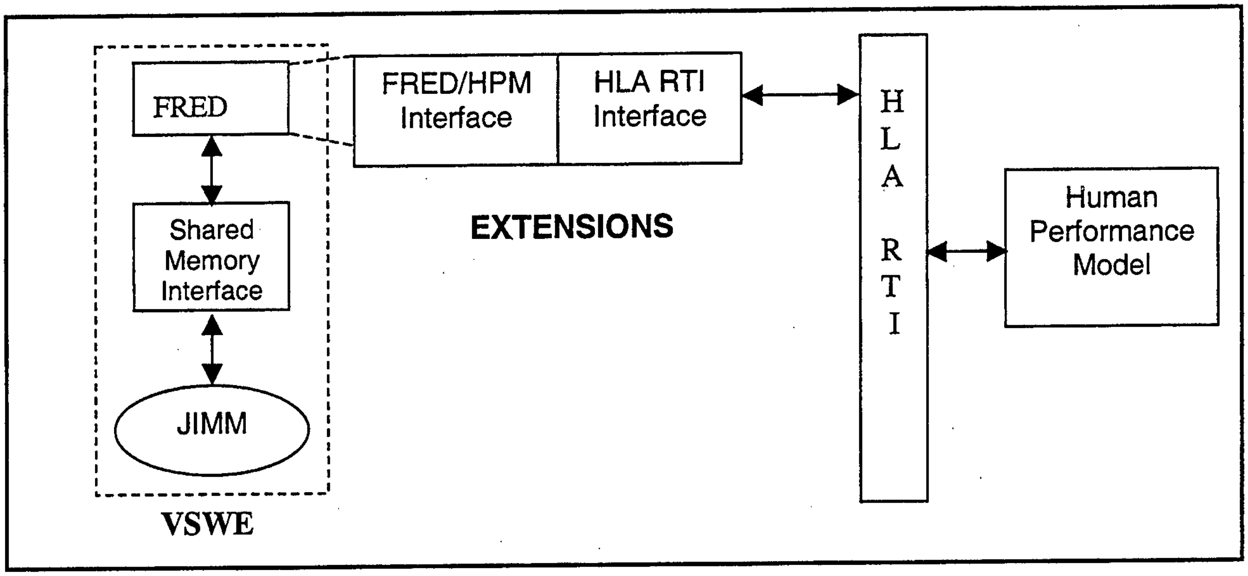


Figure 11. Simple Depiction of the Integrated Case Study 1 Simulation Environment

A minor modification to the FRED consisted of configuring the FRED Executive to run in hold-off mode with JIMM as a slave. By using a hold-off mode, the CART task can time step the FRED when it is allowed to run (e.g., when a time advance grant is received), allowing the constructive simulation to remain synchronized with the HPM. In addition, the Mission Computer software was modified slightly to include changes to the Auto-Pilot Sequencing Logic, the addition of a Move Cursor procedure that did not latch to nearest object, and added logic for the Fly-to-Heading and Commanded Airspeed functions. Finally, two minor modifications were made to JIMM. First, the memory allocations for JIMM were changed to increase JIMM reliability. Second, JIMM was changed to run as a slave to the FRED Executive. Through this mechanism, the FRED tells JIMM when to freeze and when to run.

This page left blank intentionally

4.0 METHOD

As described earlier, Case Study 1 was conducted within the VSWE environment established by the JSF program for their air-to-ground simulation efforts. Using the VSWE TCT mission described earlier, two sets of data collection trials were conducted. The first set consisted of trials in which the HPM controlled the aircraft simulation via HLA (the 'HPM condition'). In the second data collection effort, eight trained pilots flew the same simulated missions as those flown by the HPM. This was called the 'Human-in-the-Loop' or 'HITL condition'. Mission data from the HPM and pilot-commanded trials were then analyzed and compared to determine whether results from HPM-controlled simulation runs approximated those from the pilot-controlled runs across various mission performance dimensions.

4.1 Case Study 1 Experimental Design

The experimental design for Case Study 1 was a between-subjects design where operator type (HPM vs. HITL) was the primary factor. *Scenario* was a dimension added to increase the variability of the data within subjects and to test the robustness of the HPM. The six scenarios used in VSWE 3B were re-used for this effort. Changes in the experimental conditions from scenario to scenario included different planned routes to the target, different pop-up threat locations, different target locations, and different target update coordinates. All of the scenario differences were minor, but they did create differences in the way each scenario played out as well as variability in operator performance. One goal was to determine whether the variability in HPM performance across these six scenarios tracked with that observed in the HITL trials.

Each pilot in the HITL condition performed six repetitions of the part-mission TCT strike scenario, one trial for each of six different scenario variations. The HPM also performed six repetitions of each scenario variation to complete the 84-cell matrix. Table 3 presents a randomized scheme for ordering subject exposure to scenarios that controlled order effects. In Table 3, a trial number filling each scenario cell denoted presentation order.

Table 3. The Trial Presentation Order For Constructive and Virtual Subjects

*Subject	Operator Type	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
1	HITL	1	5	3	6	2	4
2	HITL	5	3	4	2	6	1
3	HITL	3	4	2	5	1	6
4	HITL	6	2	1	4	5	3
5	HITL	2	1	6	3	4	5
6	HITL	4	6	5	1	3	2
7	HITL	4	2	6	1	5	3
8	HITL	3	5	6	2	4	1
9	HPM	3	5	6	2	1	4
10	HPM	6	3	1	4	5	2
11	HPM	1	2	6	3	4	5
12	HPM	5	1	3	4	2	6
13	HPM	2	4	1	5	6	3
14	HPM	3	6	5	2	1	4

*Subjects 1-8 represent the 8 pilots in the HITL condition. 'Subjects' 9-14 represent 6 blocks of HPM trials, each treated in the analysis as a subject.

4.2 HITL Testing

4.2.1 Participants

Eight subjects, all with military, tactical fighter pilot experience took part in the HITL test during September and October 2000. Seven of the pilots were USAF reservists while one was a member of the Ohio Air National Guard. All of the participants were qualified as senior pilots (at least 1500 flight hours) or command pilots (at least 3000 flight hours). All of the pilots had experience in multiple single-seat aircraft, but their most recent experience was in either F-4, F-16, or A-10 aircraft. Five of the eight had experience in the F-16, which was critical given the similarity between the simulator hands-on throttle and stick (HOTAS) and that of the F-16.

4.2.2 Apparatus

The apparatus for the HITL testing consisted of the FRED cockpit and MICS used in the VSWE environment described earlier in section 2.2.1.

4.2.3 Procedure

During the HITL phase of the study, data collection was organized into four sessions -- with two pilots in each session. Each session included a training phase followed by a data collection phase. The training phase consisted of an overview briefing followed by familiarization training in the cockpit. After the training phase was complete, each pilot flew six data collection trials with the order of the trials randomly determined to minimize order effects.

4.2.3.1 Training Phase. The training phase began with an overview briefing. The overview covered the purpose of the study, the types of missions the pilots were expected to conduct, and the tactics that were to be employed during various phases of the missions. Topics addressed in this briefing included the Study Purpose and Objectives, the Scenario Threat/Target Data, Planned Routes, and Target Acquisition and Threat Avoidance Tactics.

After a short question and answer period, the pilots were immersed in the cockpit for the familiarization training. Familiarization training reviewed interaction with displays and controls in the cockpit, focusing on only those features and capabilities that were needed for the study (i.e., HOTAS usage, sensor manipulation, re-planner usage). During this segment of training, both pilots were seated in a cockpit and a trainer led each of them through a series of exercises designed to demonstrate the VSWE cockpit switchology. These exercises included switching between displays of interest, activating the different modes of the radar and infrared sensors, observing moving target indications, slewing the cursor and designating desired points of interest, interacting with the in-flight re-planner, and entering data into the UFC panel. The pilots were then free to ask questions and experiment with control and display interaction.

Once both pilots and the trainers were satisfied that their knowledge of the HOTAS and HDD display interaction was adequate, they graduated to part-task rehearsal. During this segment of their training, each pilot rehearsed sensor management activities and evasion tactics within small segments of a VSWE 3B unit interdiction (UI) mission. The UI mission was similar to the TCT mission, except that the target was an armored column. One of the

elements in the armored column was replaced with a SCUD to expose pilots to the intended target of the TCT scenarios.

For sensor management, each pilot was instructed to employ the sensors to detect moving targets, add moving target indications to a shootlist, and then use the combination of sensors to detect, identify, and designate a SCUD target. For evasion, each pilot was instructed to fly a preplanned route in autopilot until the pilot detected a threat launch. After launch detection the pilot was to determine the threat type and implement a course of action based on the threat type. If the threat required evasive action, the pilot was to disengage the autopilot, release a countermeasure series, and turn to 180 degrees away from the launch indication at full afterburner until the threat launch indication subsided. These activities were practiced a few times until both pilots and the trainers were satisfied that the pilots had reached an appropriate level of proficiency for each part-task.

Finally, each pilot integrated the elements of the part-task rehearsal into a full mission rehearsal exercise using the same UI mission that was used during part-task rehearsal. During the full-mission rehearsal each pilot flew a full mission similar to those that he experienced during data collection, forcing him to implement most or all of the part-task activities in an integrated fashion. Once the trainers and the pilots had determined that each pilot was proficient in the conduct of the integrated strike mission, the training phase was deemed complete. Upon completion of the practice trials, each subject performed the six data collection trials. The order of these trials was counter-balanced across subjects. The duration of each trial was approximately 30 minutes real time.

4.2.3.2 Data Collection Phase. During data collection, each pilot/cockpit had a test director and a test conductor. The test director was situated with the pilot/cockpit in the room housing both cockpits while each test conductor was situated in the SIMAF battle room. The test director was responsible for providing instruction to the pilot, answering pilot questions, and coordinating the selection of the trial scenario. The test conductor was responsible for executing the FRED and JIMM software for that pilot/cockpit and ensuring that the data collection files were archived appropriately prior to the next run. Communication between each test director and test conductor was maintained via the intercom system.

On each data collection trial, the test director and test conductor for each pilot/cockpit coordinated their efforts so that the appropriate scenario variation was flown for that pilot/cockpit. Each trial began with the aircraft in flight approaching the acquisition legs of the TCT mission. The mission computer was pre-loaded with the planned route, known threat data, and the expected target location. As the pilot followed the route toward the expected target location, he began to search the target area with the onboard sensors in an effort to acquire the TCT. The general strategy used by the pilots was to employ the real beam radar and GMTI to detect moving objects in the target area. Once these objects were detected, the pilots added them to the shootlist for later examination with the TIR. At a given time during ingress, the pilot received updated target coordinates from a simulated off-board source. He then updated the mission computer to specify the new estimated target location and began to focus imaging activities on the new location. The pilot continued to attempt acquisition until he located and identified the target or aborted the mission due to failure of target identification after a reflight of the expected target area. If and when the TCT was identified, the pilot attempted to designate it for attack, fly the aircraft to the release point, release the weapon, and egress from the target area.

In addition to the known threats, a pop-up threat was present along each route. The surface-to-air threat, which was not accounted for in the planned route, could and did launch missiles against the aircraft, forcing the pilot to take evasive action. If the pilot determined that the launched missile posed a threat to the aircraft, the pilot performed evasive maneuvers and employed countermeasures in an effort to defeat the missile.

Each trial also offered a number of opportunities for in-flight re-planning. These included recapturing the planned route after threat evasion maneuvers, planning to refly the target area if the target was not found after the first pass, planning a direct route to the target once it had been found, and aborting (returning to base) after attacking the target or after two failed attempts to find it. Re-planning activities included both acceptance of *automated* re-plans (i.e., the onboard re-planner automatically offered a route which the pilot could accept or reject), and creation of *manual* re-plans for which the pilot manually inserted desired waypoints and then requested a new plan that included the specified points. Pilots were

instructed to accept all automated re-plans when offered, as the onboard re-planner was designed to offer an optimal route. Manual re-planning was necessary when a change to the destination was required. Manual re-plans were necessary for flying to the identified target (if it was too far off the planned route) and for reflying the target area. Pilots had to refly the target area, if after the first pass, the target had not been identified. In this case, the pilots were instructed to set up a new acquisition leg starting approximately 30 NM outside the target area, fly to the initial waypoint of that leg, turn inbound to the target area, and resume acquisition activities.

Each trial terminated once the aircraft successfully egressed from the target area or the pilot aborted the mission after two overflights of the target area without positive target acquisition. If the aircraft was intercepted by a ground threat during the course of a mission, the missile intercept was recorded, but the mission continued until the pilot initiated a mission abort after attacking the target or initiated an abort due to a failure to find the target during each of the allotted two passes over the target area.

4.3 HPM Testing

4.3.1 Participants

Constructive 'subjects' were mapped to the six trials conducted within each scenario. The mapping was accomplished by applying a randomized, without-replacement trial order.

4.3.2 Apparatus

4.3.2.1 Hardware. The computer hardware that hosted the constructive environment and the CART HPM included an Intel processor-based PC with a dual-processor 400 MHz Pentium II CPU with 256 MB RAM running the Windows NT version 4.0 (Service Pack 5) operating system, and a Silicon Graphics Octane workstation with two 300 MHz R12000 processors and 1 GB RAM running the IRIX version 6.5 Operating system. These two computers were networked through 100-Base-T networking cards.

4.3.2.2 CART HPM Environment Software. The CART HPM software version 1.05G was hosted on the PC. The CART HPM software is based on the Improved Performance Research Integration Tool (IMPRINT), a human-performance modeling environment developed by the Army Research Laboratory's Human Research and Engineering Directorate (ARL/HRED). IMPRINT is Government-owned software consisting of a set of automated aids to assist analysts in conducting human performance analyses; it provided the basic structure and modeling methodology for developing human performance models. The CART HPM software included a set of extensions to IMPRINT that enhanced its capabilities. The extensions mainly consisted of an added capability for inter-model/simulation communication via the HLA RTI and the addition of a goal orientation capability that enables human performance modelers to represent the adaptive, goal-oriented nature of human performance. The HPM communicated with other external models via the HLA RTI version 1.3 and took advantage of the RPR FOM Version 0.4/0.5 (DRAFT) implemented in the MÄK Technologies' VR-LINK 3.3 product.

4.3.2.3 FRED/JIMM Software. As in the HITL simulation trials, JIMM provided the simulation environment. However, the cockpit software environment was scaled back to incorporate only those components required for use with the HPM. One such component was the FRED software, which provided the basic representation of the aircraft and pilot interfaces. Additional capabilities were added to the FRED, enabling it to pass state data to the HPM and to reflect control inputs received from the HPM. A second capability retained for the HPM trials was the RTMP system, which was employed to pass in-flight re-planning data to the HPM. Within the HPM, some sensory processes that operated on certain displays were modeled without the actual use of those displays. For example, the probabilities of target detection and identification were calculated as functions of sensor and sensor field of view (FOV) selected, range, the location imaged, and the size/type of object within the FOV. Thus, the image generation capability of the Camber radar and the targeting infrared system were not employed.

4.3.2.4 Interface Support Software. The interface support software acted as an intermediary between the FRED/JIMM and the HPM. It obtained the current aircraft simulation data,

passed those data to the HPM, and then updated the FRED/JIMM with information as commanded by the HPM. The data exchanges occurred via HLA.

4.3.3 Procedure

In the HPM conditions, the HPM was substituted for the pilot in the loop, processing data from the mission environment and commanding inputs to the simulated aircraft. Trials were terminated according to the same conditions mentioned for the HITL trials. As in the HITL condition, one trial was conducted for each of the six scenarios. This process was repeated six times to represent the six different instantiations of the HPM shown in Table 3. While order effects were not expected from the HPM, to maintain statistical validity, the actual trials were randomly selected from a randomized block of trials without repetition and were assigned subject numbers within scenario.

4.4 Data Collection and Analysis

4.4.1 Measures of Effectiveness

The dependent variables for Case Study 1 are listed in Table 4. These measures evaluate performance on the generalized functions that drove specification of goals. Originally, a much broader set of measures had been specified. Measures involving performance times had been of particular interest. These had included the time taken to acquire and attack targets, time to react to threat launches, and time spent re-planning. Problems with recording time of events within JIMM prevented calculation of reaction to launch and attack time data. Also, lack of reliable events for collecting target detection and identification performance by pilots in the HITL condition precluded the ability to assess acquisition times.

Beyond problems with the data collection, many of the data generated by the testbed were classified. While performance measures other than those listed in Table 4 could be calculated, our ability to report them is limited. The measures in Table 4 were selected because (1) they represent a level of assessment that would be of interest to an acquisition program office and, (2) they represent performances that were judged to be relatively

independent and not subject to obvious inter-relationships that could cloud statistical interpretation of the results.

Because understanding of some of the measures require an understanding of some of the JSF sub-systems, each measure will be explained briefly here.

- <u># re-plans generated based on navigation error</u>. Because most flight control in the JSF is performed by the autopilot, navigation activity by the pilot centers on interaction with the auto-router. When the pilot deviates more than a prescribed distance (one mile for this study) off the planned route, the auto-router will compute an adjusted route and offer it to the pilot for acceptance. In the TCT scenario, this situation would occur when the pilot had to evade a SAM launch. Because navigation re-plans were triggered by pilot route deviation behavior, the number of navigation re-plans generated provided some insight into the relative comparability of route deviation between pilots and the model.
- <u># re-plans accepted based on navigation error</u>. The operator has the option of accepting or rejecting a re-plan offered by the auto-router. Because the auto-router was considered to be a key enabling technology in the JSF, the operational concept was adopted that the pilot would always accept a re-plan offered by the auto-router. This behavior was programmed into the HPM. In HITL training, pilots were instructed to always accept plans offered by the auto-router. The only exception to this procedure would be in those instances in which a pilot was engaged in evasion maneuvers and a re-plan was offered.
- <u># re-plans generated based on threat</u>. The auto-router used an electronic order of battle (EOB) database in the mission computer to predict the location of threats and then used data from the onboard threat assessment system to evaluate the activity of those threats. Based on activity of a known threat near the planned route, the auto-router might determine that adjustments to the route were necessary and offer a re-plan around the

threat. Generation of threat-based re-plans was not as directly driven by pilot behavior as navigation-based re-plans. However, large differences between pilots and the HPM in the number of threat-based re-plans could suggest that, as scenarios unfolded, significantly different routing resulted that led to different exposure to threats -- and perhaps even different opportunities to acquire the target.

- # re-plans accepted based on threat. As with navigation re-plans, the operational concept was to accept all threat-based re-plans offered by the auto-router. This provided an opportunity to determine whether pilots followed the operational concept as consistently as the HPM.
- # threat locks on ownship. A threat 'lock' occurred when an enemy SAM was able to acquire the JSF with an acquisition radar. This measure counted the number of times locks occurred during a trial. A given SAM could lock on the JSF more than once.
- # threat launches at ownship. This measure counted the number of missiles that were launched at the JSF during a trial.
- % of threat missiles defeated by ownship. A missile was defeated when it did not achieve a kill on the JSF. There were a variety of reasons a missile could be defeated. Track could be lost on the JSF. The missile could run out of fuel. The missile could get close to the JSF and detonate, but damage assessment algorithms could determine that the JSF was not destroyed. This measure allowed us to assess the relative effectiveness of pilots and the HPM at evading threats.
- Probability of correct acquisition. Ultimately, target acquisition involved the process of acquiring the Theater Ballistic Missile (TBM) target with the TIR and designating it to the weapon system. This measure assessed the ability of pilots and the HPM to routinely and correctly acquire the TBM (as opposed to some other object such as a tank, or nothing at all).
- Range at weapon release. A launch acceptance region (LAR) was provided on the TSD as graphical symbol that showed the effective

engagement envelope of the given weapon at the current speed and altitude. A simple indication of the consistency with which LAR guidance was followed was the range at release.

- Probability of kill given attack. This was another evaluation of attack effectiveness that assessed whether a kill occurred given an attack was conducted.

Table 4. Dependent Variables Collected During Case Study 1

Mission Function	Measure of Performance
Navigation	# re-plans generated based on navigation error†
	# re-plans accepted based on navigation error†
	# re-plans generated based on threat†
	# re-plans accepted based on threat†
Threat Evasion	# threat locks on ownship†
	# threat launches at ownship†
	% of threat missiles defeated by ownship†
Target Acquisition	Probability of correct acquisition
Target Attack	Range at weapon release†
	Probability of kill given attack

† Dependent variables used in the statistical analyses (see Section 5.1 and Table 5).

4.4.2 Data Sources

The data sources included binary data files generated by the FRED and JIMM during all trials, and were supplemented with files generated by the CART HPM during constructive runs. The data captured in the FRED files were event-based data, such as RTMP interactions (i.e., re-plan requests, re-plan accepts), control and display interactions (e.g., display formats selected, targets added to the shootlist, weapon releases, etc.) and time-based data about aircraft position, attitude, and velocity, which were collected at a 5 Hz rate. Each event,

whether it was time-based or event-based was time stamped with a time value common to all data sources.

The JIMM data were captured in a similar manner, but they tended to represent object-to-object interactions, such as threat launches, radar tracks, weapon releases, and target kills. The CART HPM data consisted of text files generated by the CART HPM software that represented the initiation, duration, and termination of the HPM goals, functions, and tasks, as well as workload measures.

After each trial was completed, all the files associated with that trial were archived according to a predetermined scheme, and were transferred to another computer for reduction and analysis. All the raw data were reduced and transformed into Microsoft Excel 97 files that were subsequently imported into a Microsoft Access 97 database as tables. These tables were then queried for aggregated, high-level MOEs and lower-level MOEs according to the MOE hierarchy and the requirements of the statistical analysis.

5.0 RESULTS

Initial review of the results of pilot and HPM performance on the ten measures listed in Table 4 revealed virtually no differences on 'probability of correct acquisition' and 'probability of kill given attack'. The HPM found and correctly identified the target on 36 out of 36 trials ($p = 1.0$) and the pilots found and correctly identified the target on 47 out of 48 trials ($p = .98$). Both pilots and the HPM killed the target on every attack. Some differences were observed on the remaining eight measures. Analyses were performed to determine whether the differences were statistically significant when the variables were considered both individually and in concert. Demonstrating whether a difference existed fell to three categories of measures as produced with the aid of Statistical Package for the Social Sciences (SPSS), release 10.0 (SPSS for Windows, 1999): (1) Measures of central tendency and dispersion, (2) inference, and (3) internal structure -- this last being derived as a byproduct of the other two. Measures of central tendency and dispersion (descriptive statistics) included means, standard deviations and intercorrelations among the dependent measures. Measures of inference included repeated measures and multivariate and doubly multivariate analyses of variance (MANOVA). Measures assessing whether internal structural differences existed between the two operator types included an index developed by the Department of Psychology at the University of Akron in the 1980's, called the *congruency ratio*. As adapted for this study, the congruency ratio was based on the intercorrelation matrices. Also included as an assessment of internal structure were tests of homogeneity of variance/covariance (Levene and Box's M tests) drawn prior to the interpretation of MANOVA.

5.1 Descriptive Statistics: Measures of Central Tendency and Dispersion

The eight dependent measures subjected to statistical analysis were all classified. A means was needed to present these data in an unclassified format that hid the true responses of the operator, yet preserved any differences between operator types. A 'mean rank' procedure was used that transformed each of the eight dependent measures onto the same numeric scale. Table H-1 of Appendix H shows an example of how this was done. This method was

used solely to provide means and standard deviations, but not the correlations. The inferential results, that in this report depict the model effects at a general level, are such that the actual data values cannot be defined. For convenience and brevity in the following discussions, an identifying number was assigned to each of the dependent variables associated with a set of ranks. Table 5 lists the dependent variables along with their identifying numbers.

Table 5. Index of Dependent Variable Identifiers to Variable Names.

Identifier	Variable Name
DV1	# re-plans generated based on navigation error
DV2	# re-plans accepted based on navigation error
DV3	# re-plans generated based on threat
DV4	# re-plans accepted based on threat
DV5	# threat locks on ownship
DV6	# threat launches at ownship
DV7	% of threat missiles defeated by ownship
DV8	Range at weapon release

Figure 12 displays the overall average mean ranks and 95% confidence intervals for the eight dependent variables of interest. Referring to Figure 12, on average, the HITL pilots showed poorer performance on several of the dependent measures; namely, higher '# threat locks on ownship', '# threat launches at ownship', and 'range at weapon release' and lower '% of threat missiles defeated by ownship'. The results were not as clear-cut with the remaining measures. For example, the pilots generated and accepted more navigation error-related re-plans, while the HPM generated and accepted more threat-based re-plans.

In regard to variability, the percentage of variation around the mean -- known as the *coefficient of variation* (standard deviation divided by the mean times 100) -- was generally less for the HPM than for the HITL. However, the percentages were greater than expected for the HPM, as one example, a HPM high of 24% compared to a HITL high of 31% for 'range at weapon release'. Even so, on one measure the reverse was true; a high of 17% for the HPM compared to a HITL high of 12% for '# threat launches at ownship'. Moreover, the

results were not consistent across all scenarios. Perhaps a clearer picture of overall variability is displayed in Figure 12. For the first four variables the length of the confidence intervals around the mean ranks are approximately the same for both the HPM and the HITL. As for the remaining variables, three – DV5, DV7 and DV8 – exhibit somewhat greater variability for the HITL than for the HPM; however, DV6 shows pronounced greater variability for the HPM. Also note that the confidence intervals for the two operator conditions overlap for all dependent variables except DV4.

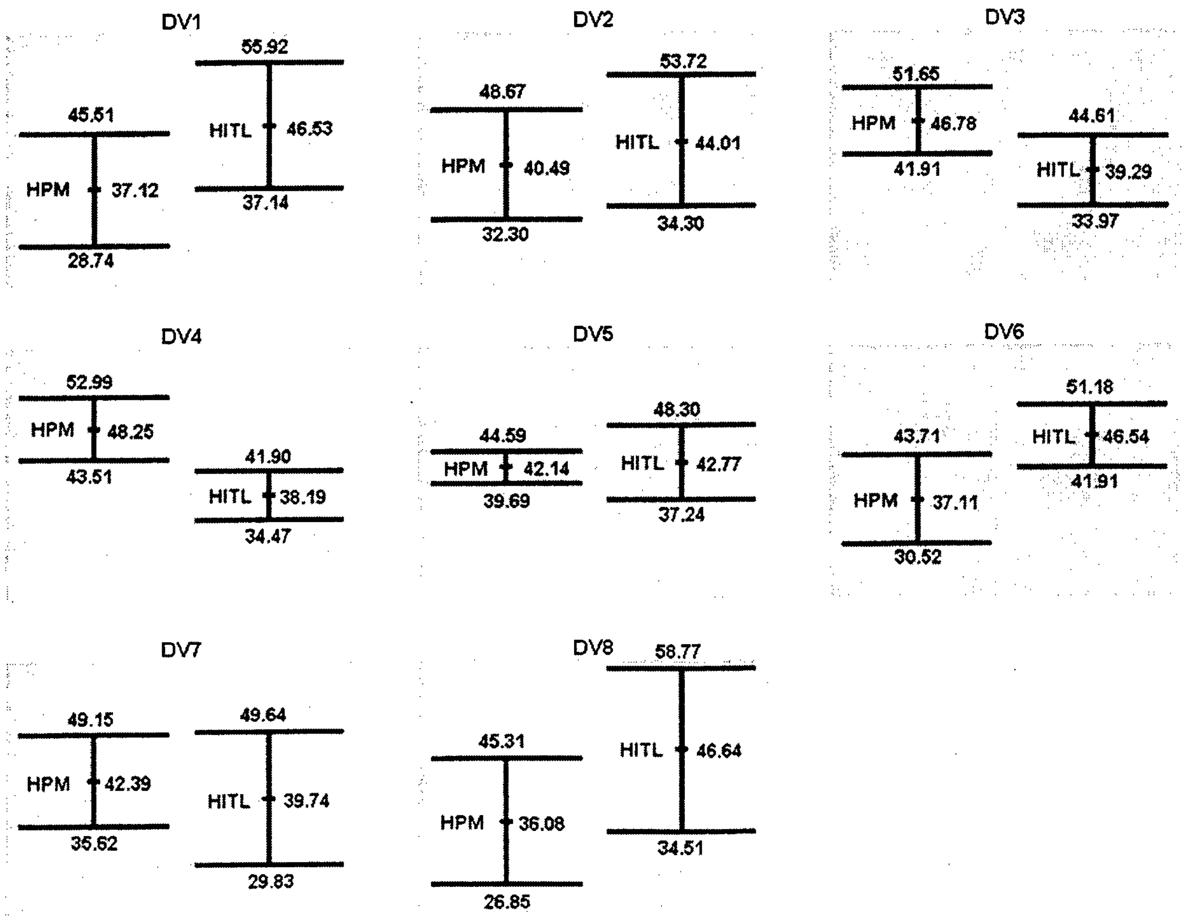


Figure 12. Comparison of Operator Types on Overall Average Ranks[2]

[2] Mean ranks and 95% Confidence Intervals for the eight dependent variables. The means are presented at the centers, with the upper and lower confidence bounds annotated on the whiskers. Data for the HPM condition are on the left of each panel, while data for the HITL condition appear on the right.

Although the scenarios were treated as equivalent, some of the scenarios revealed results opposite to the overall average. For example, in scenarios one and two, the HPM demonstrated a greater number of threat locks and launches on ownship (mean rank of 43.08 and 34.17, respectively, for scenario one, and 67.92 and 67.58, respectively, for scenario two) than did the pilots (mean rank of 38.06 and 31.81, respectively, for scenario one, and 33.81 and 48, respectively, for scenario two). For scenario one, the pilots defeated a greater percentage of missiles than did the HPM. The range at release was further for the HPM than the pilots in scenario three. Finally, the HPM generated and accepted more navigation-related re-plans than did the pilots for both scenarios two and four. Further detail for the descriptive statistics can be found in Appendix I, Table I-1. Note that measures of central tendency and dispersion for ranked data typically involve medians and interquartile deviations. In this instance, however, because the classified source data would be described by means and standard deviations, it was deemed appropriate to treat the ranks that represent those data similarly.

5.1.1 Descriptive Statistics: Correlations Among the Dependent Variables

In Appendix J, Tables J-1 and J-2 display the correlations among the dependent measures for the total sample and by operator type, respectively. These correlations are based on the original classified data. Intercorrelations based upon the total sample (N=84 for the correlations among DV1 to DV6, N=81 for correlations with DV7, and N=83 for correlations with DV8 due to missing data) provided the framework for determining the multivariate models explained in the next section. The intercorrelations by operator type also became important for a subsequent procedure, the calculation of the *congruency ratio* as discussed in a later section. What is important to note here is that, in the majority of cases, the correlations among the dependent variables for HPM exhibited similar strength and the same direction as the correlations for HITL. However, there were several exceptions. DV3 and DV4 correlated with all the other variables exactly the same for the HPM condition, while there was some -- though not excessive -- variation for the pilot condition; this was also true of DV1 and DV2. There were some instances of correlations of dissimilar strength. For example, the HPM number of navigation error-related re-plans generated and accepted (DV1

and DV2) had highly positive and significant ($p<.01$) correlations with the number of launches against ownship (DV6). The respective correlations for the pilots were still positive and significant, but to a much lesser degree. Additionally, the number of locks and launches (DV5 and DV6) inversely correlated with range at release (DV8) for the HPM to a greater degree than for the pilots. There were differences in sign for several variables, i.e., one correlation was positive while the other was negative; however, with two exceptions, neither of the correlations was significant ($p<05$). Both number of threat-based re-plans generated and accepted (DV3 and DV4) showed a significant ($p<.01$), negative correlation with percent missiles defeated (DV7) for the HPM, while the respective correlations for the pilots were nonsignificant ($p>.05$) and positive. We do not have sufficient data to explain these correlation anomalies. They may have been due to genuine internal differences between the two operator conditions or simply an artifact of the simulation environment or the experiment.

5.2 Measures of Inference

5.2.1 Multivariate Analyses

HPM task times were held constant, so the variability in the HPM condition across trials was due solely to the variability arising from the stochastic nature of the simulated environment.[3] Variability in pilot performance from trial to trial contributed some additional variability in the HITL condition. The multivariate repeated measures analyses that were used in making inferences control for this additional source of variability in the HITL condition. Performing this analysis required examining beforehand the matrix of correlations among the dependent variables, finding those sets of variables that correlate with each other significantly ($p<.05$), and forming models to simultaneously test the sets of variables in what is known as a 'doubly multivariate' analysis.

[3] Although the human performance task–network modeling environment provides a convenient means for introducing variability into task times, this was not done because there were no available data regarding task variability, and any task variability simulated would have been based upon conjecture. The confidence intervals presented in Figure 12 do not suggest that overall variability was markedly different between the HPM and the HITL conditions.

The following example uses five dependent variables to illustrate the concept -- where '↔' is used to indicate a significant correlation (at $p<.05$):

DV1↔DV2	DV2↔DV3	DV3↔DV4	DV4↔DV5
DV1↔DV3		DV3↔DV5	
DV1↔DV5			

From these intercorrelations, three models would result -- (1) DV1, DV2, DV3; (2) DV1, DV3, DV5; and (3) DV3, DV4, DV5. The technique used to derive the models in the example is relatively straightforward. DV1 correlates significantly with all variables except DV4 -- but a four-variable model is not possible since DV2 does not correlate with DV5; thus DV1 exists in two models. The second model with DV1 results from the fact that DV3 correlates with DV5, and both correlate with DV1. That leaves DV4 unassigned. If DV4 did not correlate with any of the other variables, a model with just DV4 would be appropriate; however, DV4 correlates with both DV3 and DV5, resulting in the third model. This concludes the example.

Examining the matrix of intercorrelations for the entire sample in the study (Table J-1, as noted in the last section) revealed three sets of variables, forming the following three doubly multivariate models:

Model 1 -- DV4, DV5, DV6, DV8;
Model 2 -- DV3, DV4, DV7;
Model 3 -- DV1, DV2, DV5, DV6

Tables 6, 7, and 8 display the results of the omnibus tests for each of the three models.

Table 6. Multivariate Tests[a] for Model 1: Effect of Between Subjects Operator Type

DV4, DV5, DV6, DV8	Value	F	Hypothesis df[b]	Error df[b]	Sig.	Partial Eta Squared	Population Eta Squared
Pillai's Trace	.748	5.924	4.000	8.000	.016	.748	.727
Wilks' Lambda	.252	5.924	4.000	8.000	.016	.748	.727
Hotelling's Trace	2.962	5.924	4.000	8.000	.016	.748	.727
Roy's Largest Root	2.962	5.924	4.000	8.000	.016	.748	.727

a. Design: Intercept+OPERTYPE; Within Subjects Design: SCENARIO
b. The totals for the degrees of freedom (df) were less than 13 due to missing data for Model 1.

Table 7. Multivariate Tests[a] for Model 2: Effect of Between Subjects Operator Type

DV3, DV4, DV7	Value	F	Hypothesis df[b]	Error df[b]	Sig.	Partial Eta Squared	Population Eta Squared
Pillai's Trace	.786	8.586	3.000	7.000	.010	.786	.764
Wilks' Lambda	.214	8.586	3.000	7.000	.010	.786	.764
Hotelling's Trace	3.680	8.586	3.000	7.000	.010	.786	.764
Roy's Largest Root	3.680	8.586	3.000	7.000	.010	.786	.764

a. Design: Intercept+OPERTYPE; Within Subjects Design: SCENARIO
b. The totals for the degrees of freedom (df) were less than 13 due to missing data for Model 2.

Table 8. Multivariate Tests[a] for Model 3: Effect of Between Subjects Operator Type

DV1, DV2, DV5, DV6	Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared	Population Eta Squared
Pillai's Trace	.736	6.260	4.000	9.000	.011	.736	.716
Wilks' Lambda	.264	6.260	4.000	9.000	.011	.736	.716
Hotelling's Trace	2.782	6.260	4.000	9.000	.011	.736	.716
Roy's Largest Root	2.782	6.260	4.000	9.000	.011	.736	.716

a. Design: Intercept+OPERTYPE; Within Subjects Design: SCENARIO

For all three models, all of the multivariate tests were significant at alpha levels less than .05. Moreover the eta-squared index values above .7 indicate a strong overall effect.[4] Post hoc tests were used to determine to what the differences found are attributable. Barker and Barker (1984) advanced Hummel and Sligo's premise, based on Monte Carlo simulations, that after executing an omnibus MANOVA test and finding significance, conducting multiple individual tests of the dependent variables will reveal where the differences exist and protect the experiment-wise alpha level (typically .05). Others have disagreed, preferring instead simultaneous test procedures, such as the Bonferroni. Bray and Maxwell have indicated that either category of post hoc tests proved appropriate to control Type I error. In any event, both categories of tests ended with the same results in this study. Tables 9, 10 and 11 display the results of the separate tests for each of the dependent variables in Model 1, Model 2 and Model 3, respectively.

Table 9. Tests of Between-Subjects Effects for the Dependent Variables of Model 1

Source	Measure	df	F	Sig.	Partial Eta Squared	Population Eta Squared
Operator Type	DV8	1	1.862	.200	.145	~0
	DV6	1	9.597	.010	.466	.421
	DV5	1	.031	.864	.003	~0
	DV4	1	13.812	.003	.557	.520
Error	DV8	11				
	DV6	11				
	DV5	11				
	DV4	11				

Table 10. Tests of Between-Subjects Effects for the Dependent Variables of Model 2

Source	Measure	df	F	Sig.	Partial Eta Squared	Population Eta Squared
Operator Type	DV3	1	4.626	.060	.340	~0
	DV4	1	21.154	.001	.702	.672
	DV7	1	.556	.475	.058	~0
Error	DV3	9				
	DV4	9				
	DV7	9				

[4] Population estimates of the strength of the effect, *eta squared*, were derived from a formula provided by Bray and Maxwell (1990).

Table 11. Tests of Between-Subjects Effects for the Dependent Variables of Model 3

Source	Measure	df	F	Sig.	Partial Eta Squared	Population Eta Squared
Operator Type	DV1	1	.405	.537	.033	~0
	DV2	1	.052	.823	.004	~0
	DV5	1	.009	.927	.001	~0
	DV6	1	9.227	.010	.435	.391
Error	DV1	12				
	DV2	12				
	DV5	12				
	DV6	12				

Looking at the three sets of tests, just two of the eight dependent measures remained significant ($p<.05$). These were DV4 (# re-plans accepted based on threat) and DV6 (# threat launches at ownship). The strength of the effect population estimates (eta squared) for DV4 ranged from .52 in Model 1 to .67 in Model 2, and for DV6, from .39 in Model 3 to .42 in Model 1. All four cases showed moderate effects, unlike the strong effects found in the omnibus tests. Note that for nonsignificant ($p<.05$) F values, the population eta squares were treated in effect as zero.

5.2.2 Outliers

SPSS defines outliers as those values greater than 150 percent of the interquartile range added to the value at the 75th percentile, or less than 150 percent of the interquartile range subtracted from the value at the 25th percentile[5]. For example, if the middle 50 percent of data values for some dependent measure ran from 60 to 90, the interquartile range would be 30 and values above 135 and below 15 would be considered outliers.

SPSS identified six percent of all the data used in the analysis as outliers. Scenarios 2 and 3 accounted for the overwhelming majority of outliers in the Case Study 1. Doing a reanalysis with just the four remaining scenarios revealed that the omnibus tests for Model 1, 2 and 3 were still significant ($p<.05$), but DV6 did not remain significant ($p>.05$) as Tables 12 and 14

[5] The interquartile range is the range between the 25th and 75th percentile values (i.e., the middle 50 percent of data values). Note that SPSS relies on "Tukey's Hinges" to determine the 25th and 75th percentile values that it uses in identifying outliers; these are usually different values than those that might be found with a common frequency analysis program.

show. However, the strength of the effect for DV4 increased from .52 to .74 in Model 1, and from .67 to .81 in Model 2 -- as shown in Table 12 and Table 13, respectively. Also, DV3 (# re-plans generated based on threat) was now significant ($p<.05$) in Model 2, due in part to a smaller mean square error arising from two thirds of the DV7's missing data having been associated with the two deleted scenarios. However, note that DV3's partial eta squared value increased twice over the original value for Model 2. Given that eta squared is an estimate based on the sum of squares, the increase cannot simply be due to a change in the error degrees of freedom.

Table 12. Tests of Between-Subjects Effects for the Dependent Variables of Model 1 Without Scenarios 2 and 3

Source	Measure	df	F	Sig.	Partial Eta Squared	Population Eta Squared
Operator Type	DV8	1	4.180	.066	.275	~0
	DV6	1	1.073	.323	.089	~0
	DV5	1	.037	.851	.003	~0
	DV4	1	34.132	.000	.756	.736
Error	DV8	11				
	DV6	11				
	DV5	11				
	DV4	11				

Table 13. Tests of Between-Subjects Effects for the Dependent Variables of Model 2 Without Scenarios 2 and 3

Source	Measure	df	F	Sig.	Partial Eta Squared	Population Eta Squared
Operator Type	DV3	1	25.579	.000	.699	.674
	DV4	1	52.634	.000	.827	.813
	DV7	1	.827	.383	.070	~0
Error	DV3	11				
	DV4	11				
	DV7	11				

Table 14. Tests of Between-Subjects Effects for the Dependent Variables of Model 3 Without Scenarios 2 and 3

Source	Measure	df	F	Sig.	Partial Eta Squared	Population Eta Squared
Operator Type	DV1	1	.122	.732	.010	~0
	DV2	1	.256	.622	.021	~0
	DV5	1	.032	.861	.003	~0
	DV6	1	1.924	.191	.138	~0
Error	DV1	12				
	DV2	12				
	DV5	12				
	DV6	12				

5.3 Measures of Internal Structure

A measure developed in the Department of Psychology at the University of Akron was adapted for use as a way of arriving at a single, descriptive metric for assessing the overall commonality of variation between the HPM and HITL conditions. This metric, known as a *congruency ratio*, affords a means to quantify the overall relationship between the two operator conditions by examining the intercorrelations among the eight dependent variables for each condition. The pattern of correlations within the HPM and HITL conditions are in effect themselves correlated via the congruency ratio -- which can be viewed as a coefficient of meta correlation. The congruency ratio (CR), as used here, is defined in Equation (1) as:

$$CR = \frac{\sum_{i=1}^{n}(A_i \bullet B_i)}{\left(\sqrt{\sum_{i=1}^{n} A_i^2} \bullet \sqrt{\sum_{i=1}^{n} B_i^2}\right)} \qquad \textbf{(1)}$$

where:

A_i corresponds to cell i of the correlation matrix of the HITL Dependent Variables

B_i corresponds to the corresponding cell i of the correlation matrix of the HPM Dependent Variables

i is summed across all cells above the principal diagonal of the correlation matrix (i.e., summed across all intercorrelations)

$n =$ j (j-1)/2, for j dependent variables (i.e., n is the total number of intercorrelations)

Using notional data, Table 15 and Equation (2) demonstrate the computation of the congruency ratio defined in Equation (1).

Table 15. Example Calculation of a Congruency Ratio (CR)

Group A	Group B	A • B	A^2	B^2
0.30	0.40	0.12	0.09	0.16
-0.20	0.20	-0.04	0.04	0.04
0.50	-0.30	-0.15	0.25	0.09
-0.30	-0.80	0.24	0.09	0.64
-0.40	0.20	-0.08	0.16	0.04
0.70	0.10	0.07	0.49	0.01
	Sums=	0.16	1.12	0.98

$$CR = \frac{\sum_{i=1}^{n}(A_i \bullet B_i)}{\left(\sqrt{\sum_{i=1}^{n} A_i^2} \bullet \sqrt{\sum_{i=1}^{n} B_i^2}\right)} = \frac{0.16}{\sqrt{1.12}\sqrt{0.98}} = 0.15 \qquad (2)$$

As Table J-2 of Appendix J shows, there existed 28 intercorrelations among the dependent measures for each operator type. The resulting value of CR for this study was 0.78. Squaring the CR provides an index called a *coefficient of determination* (Neter & Wasserman, 1974) which, in this case, is the amount of variability in the HITL condition's structural pattern of correlations that was explained by the HPM condition's structural pattern of correlations. In other words, the HPM accounted for 61 percent of the variation in the pilots' behavior in the HITL condition.

Another way of assessing structural differences between the operator types is through examination of the variance/covariance matrices. The Levene test of homogeneity of variance is an SPSS option available for analysis of the MANOVA models. Levene's test, a univariate procedure, compared the variance exhibited by the model with that exhibited by the pilots for each scenario within each dependent variable separately. For example, the HPM variance in scenario one for DV1 was compared to the HITL variance in scenario one

for DV1. For those tests where the index was found to be not significant (p>.05), equality or homogeneity of variance was indicated. Table 16 lists the scenarios passing (i.e., exhibiting nonsignificance) Levene's test.

Table 16. Results of Levene's Test

Dependent Variables	Scenarios Exhibiting Homogeneity of Variance Between Operator Types
DV1	2, 4, 6
DV2	2, 4
DV3	2, 4, 6
DV4	2, 6
DV5	4, 5, 6
DV6	2, 4, 5, 6
DV7	1, 2, 4, 5, 6
DV8	1, 2, 4, 6

As can be seen in Table 16, a number of scenarios failed the Levene test -- as many as four (DV2 and DV4) and few as one (DV7). Note that scenario three failed for all eight dependent variables. This scenario (and scenario two, but to a lesser extent) had the greatest concentration of outliers.

The Box M test of homogeneity of covariance is another SPSS option available for analysis of the MANOVA models. Box M includes a test of variance as well as extending to a test of covariance of pairs of scenarios for all dependent variables in a multivariate model. Although the scenarios were treated as equivalent (given the lack of experimental control over them), differences in the way each pilot or model iteration responded to the individual scenarios made calculation of the Box M test problematic (i.e., there was a lack of nonsingular matrices) for the three doubly multivariate models. Separate Box M tests for each dependent variable were possible only when considering just the scenarios that passed Levene's test. The results appear in Table 17.

Table 17. Results of Box M Tests

Dependent Variables	Tests of Homogeneity of Variance/Covariance Between Operator Types
DV1	Test not possible
DV2	Box M=2.091; F=0.566, p>.05
DV3	Test not possible
DV4	Test not possible
DV5	Box M=6.291; F=0.746, p>.05
DV6	Box M=34.979; F=2.133, p<.05
DV7	Box M=27.941; F=0.888, p>.05
DV8	Box M=17.627; F=1.033, p>.05

Four of the dependent variables passed the Box M test, showing equal variances and covariances between the model and the pilot. Of the three variables implicated in operator differences by at least one previous analysis, DV6 failed and -- due to the lack of nonsingular matrices -- DV3 and DV4 had no Box M test available.

5.4 Summary of Statistical Results

Three categories of measures were used to assess whether the HPM operated in the same or similar manner as the HITL. Although the measures of central tendency and dispersion evinced some differences, HPM coefficients of variation were similar to those of the HITL condition in many cases. For the two dependent variables where the inferential post hoc statistical tests (with all scenarios included) found significance, coefficients of variation were 9% versus 12% with DV4 and 17% versus 12% with DV6, for the HPM and HITL conditions respectively. For DV3, significance was observed when scenarios two and three were deleted; coefficients of variation were 10% versus 16% with DV3, for the HPM and HITL conditions respectively. Further, the measure of internal structure used to assess commonality among all the dependent measures revealed good consonance between the model and the pilots, with the model accounting for 61% of the variation in the pilots' behavior.

Interaction between operator type and scenarios was not posited as part of the hypothesis tested, but evidence for differences between scenarios and the way the two groups operated

within a particular scenario could not be ignored -- as evidenced by the tests of homogeneity of variance/covariance. The three doubly multivariate models were tested on each scenario separately. For Model 1, the omnibus tests resulted in no significance ($p>.05$) for any of the scenarios. For Model 2, significance ($p<.05$) was found for scenarios three, four and five (population eta-squares of .637, .346 and .411, respectively). For Model 3, significance was found for scenarios two, three and four (population eta-squares of .395, .474, and .527, respectively). Understanding or categorizing the scenarios would be useful for future case studies.

This page left blank intentionally

6.0 DISCUSSION

6.1 Assessment of Overall Validity of the Model

Overall the results of the descriptive and statistical analyses indicate a high degree of consistency between the performance of the HPM and that of actual pilots. This is particularly true regarding measures of performance at the *goal* level. In terms of the *Acquire Target* and *Attack Target* goals, there was little or no difference in the probability that the target would be located, correctly identified, and destroyed. Differences were observed in measures of *Navigate* and *Evade Threats* goal performance. These appear to be driven by differences in the way actual pilots and the HPM used the auto-router. Employment doctrine used in the trials dictated that -- with the exception of maneuvering to evade a launched missile -- the auto-router was to be used exclusively for making route changes. The human performance model followed this doctrine completely. Pilots were less consistent.

Observation of the HITL trials and comments from the pilots indicate that the inconsistent use of the auto-router by pilots can be attributed to quirks in the performance of the auto-router. The auto-router was a beta version of the capability. Sometimes it created routes that seemed to be counterintuitive to the situation. A good example of this are routes generated for target attack. Targets were always identified using the TIR. Because of the limited range of the TIR, the aircraft was close to the target when the identification and subsequent designation was made. Once the target had been designated, pilots were to use the auto-router to create an attack route to the target. Often, this route was not a straight line or short path to the target. Indeed a relatively long route might be generated that spiraled around the target, eventually passing over it. In these instances, some pilots would ignore the route and drive directly to the target and execute the attack. Since the beta version of the auto-router did not provide the capability to update its database with a pop-up threat's location, pop-up threats created situations that could generate unusual routes. In these cases (particularly in scenario 3), the pilot or HPM would successfully evade a threat, but once the evasion was complete, the auto-router might then offer a route that took the aircraft either back over or close to the threat so that another evasion maneuver was required (this could occur if the pop-up threat was very close to the original flight path and close to a 'must fly' point through which the aircraft had to pass). In trials flown by the HPM, this

process might be repeated several times until, eventually, the aircraft 'cleared' the threat. Pilots, on the other` hand, would immediately understand the problem and manually fly the aircraft around the threat until a route would be generated that would not overfly the threat.

While agreement in performance of the HPM and pilots was basically good at the 'goal' level, it would be appealing to know the correlation of performance at lower levels. Unfortunately, this has proven difficult. The challenge is in obtaining insight into the lower level performance of the live pilots. The most interesting aspects of operator performance are those cognitive and perceptual activities that are tied to mission success (e.g., target detection and identification, shoot list prioritization). These events are not directly observable and are difficult, if not impossible, to infer from the overt actions that are collected. An option is to pause a HITL trial and question a pilot regarding factors such as situation awareness, decision-making, workload, current goals, etc. However, this can disrupt performance. In the case study, a conscious decision was made to not disrupt the pilot and live with the loss of data. The assumption was that agreement in performance at the goal level would be sufficient because this is the level at which most decision-makers would be most concerned.

The differences between the performance of the HPM and the actual pilots that were observed raise some other interesting issues regarding model validity. At first blush, it might seem that HPM was deficient because it did not predict situations in which operators would not use the auto-router. Indeed, using data from the HITL runs, it would be possible to modify the HPM so it behaved more consistently with actual pilots. But -- is this a more valid model? It would seem that the goal of human performance modeling within the acquisition process is to predict the performance of well-trained operators, consistently applying well-defined employment doctrine and tactics. For stealthy aircraft such as JSF, technology such as the auto-router can be crucial for creating a survivable system. When testing the system, the auto-router needs to be applied consistently so that its effectiveness can be understood clearly. A real-world problem, however, is that technologies that get tested early in acquisition are not always mature. They have flaws and quirks like the auto-router used in the case study. Real operators are likely to respond to these flaws by using the technology inconsistently and, in the process, produce data that clouds the evaluation of that technology. On the other hand, human performance models can be developed that are completely consistent in their use of system capabilities and that produce data

that support a cleaner evaluation of those technologies. It is noteworthy that in this case study, the human performance model demonstrated better overall survival rates and survival-related performance (e.g., fewer launches) than the actual operators. These results suggest that even the flawed auto-router, when used consistently, enhanced aircraft survival. It would seem reasonable to conclude that the auto-router technology has merit and is worth pursuing as part of the acquisition program.

6.2 Additional Benefits Gained from Human Performance Modeling

Beyond providing a realistic representation of human performance, experience from the case study suggests that the *model development process* -- as well as the model ultimately developed -- can provide the human system team and the broader system engineering team with other benefits as well. One of these benefits is insight into effective tactics for system employment. Even though it was developed based on input from subject matter experts, the initial implementation of the strike fighter pilot model proved to be fairly ineffective at finding the time-critical target. The problem was that initial efforts to detect the target used SAR patches whose resolution was too gross to yield a visually detectable return. The model development team took a step back and re-thought the entire target acquisition process. An integrated target acquisition strategy was developed in which the GMTI and medium and low-resolution SAR images were used to for initial target detection at distances beyond effective TIR range. During this phase, a list of potential targets was developed. When the aircraft was within range, the TIR was applied to examine the potential targets and find and identify the target. As evidenced by the performance of the model, these tactics for target acquisition proved to be very effective. Their effectiveness was further validated when pilots in the HITL trials were taught the same tactics and, subsequently, exhibited target acquisition performance very close to that of the model. The conclusion of the model development team was that human performance modeling could provide a useful context for developing tactics that most effectively employ a new system or technology.

Another benefit of human performance modeling is that it provides model developers and users with an intimate understanding of the performance required of the operator. This understanding can lead to insights such as more effective function allocation between operators or opportunities

for effectively applying automation or job aiding to support the operator. In developing and using the JSF pilot model, for example, it quickly became clear that the most difficult, task-intensive portion of the job was target acquisition. Within target acquisition much time was spent performing the switch and control manipulation activities required to operate the sensors. The team realized quickly that employment of the TIR especially was driven by the shoot-list and that this manipulation was repetitious and could be automated easily. The basic concept of this automation is that:

1. As the pilot builds the shoot-list, an intelligent agent could observe the location of the airplane and the location of points on the ground of interest and compute range to those points.

2. When the aircraft comes into TIR range of a point, the agent could extend the TIR, command it to look at the point of interest and generate and save a medium and high-resolution image of the point.

3. The pilot could enter an exploitation mode on an MPD and page through the resulting imagery looking for the target, and once found, designate it for attack.

It is expected that this capability would significantly increase the number of potential targets that could be examined in a pass through the target area because much of the sensor manipulation activity would be off-loaded from the pilot. Also, it would be fairly easy to build a working prototype of the capability by reusing a portion of the JSF pilot model *Acquire Target* task network. Indeed, an important insight gained in the first case study is that a human performance model can become the basis for demonstrating an automated or aiding capability.

6.3 Implications for How M&S is Applied

6.3.1 The Challenge of Obtaining Constructive System Representations.

A key concept in CART is that integrating human performance models with constructive system and mission environments provides an opportunity to assess how operator performance can impact broader system and mission performance. Consequently, a critical ingredient for CART success is availability of constructive system models that are sensitive to variation in the performance of a CART operator model. Our experience in this Case Study suggests that such constructive system models can be difficult to obtain and that CART and other HBR modelers might have to seek alternatives.

Early in the case study, existing, available constructive models were reviewed in an effort to identify any that might be able to interact with a strike fighter pilot model. This review determined that mission level models such as Suppressor offer the ability to model a complete range of mission functions. However, the level at which actions and events within those functions are modeled is not sufficiently detailed. SAM engagements and associated outcomes, for example, are modeled using probability tables. The ability to have the aircraft interact dynamically with the threat (e.g., maneuver the aircraft, apply countermeasures) does not exist. Thus, it would not be possible to have an operator model control a Suppressor model and play out the effects of operator decision-making regarding threat evasion. On the other hand, engagement level models such as Radar Directed Gun Simulation (RADGUNS) and Enhanced Surface-to-Air Missile Simulation (ESAMS) play out the aircraft engagement by missiles and guns in great detail. But, that is all they do. They do not address target acquisition, attack or other important functions. Their scope is too narrow for the CART JSF system-modeling requirement. At the end of the review, the team concluded there were no suitable constructive system models among the existing constructive model set, and an alternative approach was required.

The alternative approach selected was to convert the virtual Mission Interactive Combat Station from the VSWE into a constructive simulation. The result was a high fidelity JSF constructive representation that was sensitive to operator model performance. Reuse of the MICS provided tremendous cost savings over developing a JSF model from scratch. Also, because the operator model did not require detailed visuals and operator station graphics, the constructive MICS could be run on lower-cost computing platforms. The original virtual MICS ran on a fourteen-processor Silicon Graphics, Inc. Onyx. The re-hosted constructive MICS runs on a two-processor Silicon Graphics Octane. There is a significant difference in cost between these two platforms. We expect that virtual simulations will provide an important source of system representation for other human performance modelers.

6.3.2 A New Paradigm for Linking Traditional Constructive and Virtual Simulation.

Traditional approaches to using modeling and simulation in system acquisition involve a mix of constructive and virtual simulation. Initially, a broad range of alternative system concepts are

defined and evaluated using constructive simulation. Alternatives that generate desired levels of performance are selected for more detailed evaluation using virtual simulation. Virtual simulations are developed which represent key capabilities of the alternatives, test plans are developed for conducting an evaluation, operators are obtained, and testing is conducted. The alternative(s) judged to be the most cost effective following virtual testing are then pushed forward into prototyping and engineering development.

While the above process offers significant improvement in the acquisition process, it still has some flaws. The constructive models used to screen system alternatives early in the process represent humans in a very limited fashion. It is difficult, if not impossible, to systematically manipulate factors of interest to human system designers. One cannot, for example, represent alternative function allocations between operators and/or machine to determine the optimal allocation scheme for a given system design. Consequently, current constructive modeling is of little use for resolving crew system design issues.

Virtual simulation provides the obvious advantage of allowing potential operators to interact with a system concept. Indeed, this is very important because the insights and information gained can be extremely valuable for crew system design. Virtual simulation, however, has its limits too. Because of the time and expense required to develop and modify virtual simulations, it often is not possible to implement the full range and combination of capabilities that might be of interest in a program. Consequently, a 'partial testing matrix' is implemented and only a subset of all possible alternatives of interest is marked for testing. The challenge here is to 'guess right' on factors such as what levels of performance of a capability should be tested, what combinations of capabilities are more important, etc.

Another limitation of virtual simulation is the human operators who participate in testing. Testing often occurs over a short period (e.g., days or weeks). For complex systems, this usually is not enough time for operators to become proficient in system employment. It is difficult to use data from these operators to predict levels of mission performance that can be achieved by highly trained, proficient operators. For new systems, a concept of employment might not exist. In this case, some portion of the test event might be devoted to letting operators 'play' with the system in order to test different tactics and operational concepts and identify those with merit. While

this provides valuable insight into effective system employment, it also reduces the time available for testing system alternatives. A secondary effect is that operators evolve their own tactics and procedures. Some operators will develop more effective tactics than others. This can lead to significant variability in performance across operators, which can, in turn, cloud the assessment and comparison of system alternatives. Finally, it can be difficult to get operators to cooperate with tactics or employment procedures in some cases. A good example of this is the auto-router used in this case study. The employment concept for the simulated JSF was that pilots would *always* use the auto-router to control the flight of the aircraft, except when under engagement by a SAM. In actual use, pilots would sometimes ignore this directive and elect to fly the aircraft themselves. This was driven, in part, by the fact that the auto-router was a beta version that sometimes produced a route that was counterintuitive. Nevertheless, pilot behavior was contrary to the procedure they were given. If the focus of the study had been to evaluate the auto-router, the pilot behavior would have made it difficult to obtain a clean assessment.

The testbed developed for the case study provides an interesting mix of capabilities that offer the flexibility and economy of constructive simulation and fidelity generally associated with virtual simulation. The result is a simulation environment that solves the problems described above and -- we believe -- offers a new paradigm for integrating constructive and virtual testing. First, a CART human performance model linked to what used to be a virtual simulator provides a rich constructive environment for exploring operator issues associated with system alternatives and concepts. This provides an acquisition program simulation team with an opportunity to resolve these issues before proceeding to virtual simulation. Also, because of its high fidelity system representation, a CART testbed provides an opportunity to screen the system alternatives identified through traditional constructive simulation, subjecting them to more intense scrutiny than ordinarily possible with constructive simulation. Problems with an alternative can be identified before taking on the time and expense of implementing it in virtual simulation. Indeed, a CART testbed can be used to conduct testing on a complete test matrix prior to conducting virtual simulation. A partial test matrix can then be applied in virtual testing in an effort to confirm results of constructive testing. Because the HPM can be driven by data produced by the models that underlie the system simulation and does not need detailed user interfaces, the cost of modifying the constructive system simulation to represent different system alternatives can be much less than modifying a virtual simulator. This permits implementation

and testing of a greater range of alternatives. A CART testbed also can be used to resolve tactics, concepts of operations, procedures and other system employment-related issues. This was demonstrated in the earlier discussion of how CART was used to derive more effective tactics for sensor employment in JSF. During virtual testing, these tactics can be taught to participating operators. This makes their performance more uniform and proficient. It also reduces the amount of time required for operators to train and 'play' with the system, leaving more time for testing. Finally, a CART HPM can be developed that faithfully applies tactics and follows employment concepts providing a clear assessment of the technology or capability of interest.

Not only is CART a feed-forward capability that bridges the gap between traditional constructive and virtual simulation, it can become a feed-back mechanism that exploits data collected in virtual simulation to improve the CART HPM and, in turn, feed data back to the traditional constructive simulations. Once the virtual simulation runs are complete, the data generated by the operators and the information gathered in post-mission debriefs can be used as appropriate to refine performance of the CART models to better reflect real operator performance. Also, data generated by a CART testbed can be used to update traditional constructive simulations. For example, a CART testbed can be used to fly vectors past different SAM sites that vary parameters such as speed, altitude, and bearing relative to the site and then implement evasive maneuver when the SAM launches. Outcomes of the engagements can be recorded and the data can be used to update SAM probability of kill tables in models such as Suppressor so the data more accurately represent effects of a pilot on SAM survival.

6.4 Assessment of the CART HPM Tool and Architecture

In general, the case study team was pleased with the IMPRINT-based task network-modeling tool. The graphical user interface made it easy to specify tasks and develop networks. Also, the graphically based tool for mapping external variables to SIMAN interactions proved easy to use and, ultimately, will provide a significant savings to CART users because they will not have to re-write the CART middleware each time a new HPM is built.

Perhaps the most powerful feature of the tool was the *goal* capability that was added as part of the CART program. As expected, the goal structure yielded a HPM that responded dynamically

to changes in the mission environment. It was interesting to observe how variations on a core scenario were able to generate significantly different model performance in terms of the number of times goals fired and the duration that goals were active. This confirms the expectation that an advantage of being able to connect human performance models to system and mission environment models is that it provides the human factors analyst and, indeed, the entire system development team with tremendous insight into how the mission environment and system design drive performance of the operator.

While the CART tool had many positive aspects, there were some limitations. One of these was the programming language inside the model development environment. One of the realizations gained in the case study was that CART models require much more programming than traditional IMPRINT models. Some of this is driven by the interface with the constructive system representation. A large number of variables that receive data (information) from the constructive system simulation and pass actions back to the simulation must be defined and managed within a CART HPM. Equally important is the need to represent cognitive, perceptual and other processes that underlie task performance. In the JSF pilot model, for example, extensive code was written that represented the operator's perception of targets on the sensor displays and determined when target detection and identification could occur. The language does not support complex conditional statements, but is limited to simple 'If-Then-Else' statements. Also, it does not support nesting of conditional statements. In order to create nested conditionals, individual conditional instruction segments are embedded in macros and one macro calls another that calls another, etc. It is an awkward arrangement that can make debugging a challenge. A powerful addition to future versions of the CART HPM development environment would be a more extensive, robust programming environment with good debugging tools.

A limitation of the overall CART testbed was speed of execution. The integrated CART/FRED/JIMM simulation ran three to four times slower than real-time. Since completing the case study, the Defense Modeling and Simulation Office (DMSO) has provided funding to explore ways to make the simulation run faster. The CART development team has optimized elements of the CART runtime environment and has been able to increase performance to about 1.8 times real-time. At this point the obstacle is the FRED/JIMM simulation, which runs at 1.3 times real-time. The remainder of the delay is driven by overhead associated the HLA RTI using

time management services and running regulated and constrained. The lesson learned here is that if real-time performance is desired in a CART simulation, all federates must be capable of achieving better than real-time performance to off-set delays imposed by the RTI. For CART HPMs themselves, this does not appear to be a problem. These models tend to run many times faster than real-time.

Beyond the tool itself, perhaps the most important insight gained is the need for efficient and effective data collection, management, and analysis tools. The CART concept for data analysis is to develop a hierarchy of performance measures and data that can be used to trace and evaluate how low-level operator performance impacts high level functions and objectives. While the data and measures for the case study could not be discussed in detail, it is the opinion of the CART team that the performance-measure hierarchy does provide an effective means for explaining operator effects. The challenge is the level of effort it takes to generate the measures. The testbed developed for the case study generates massive amounts of data from the JIMM mission environment, the MICS system representation, and the CART HPM. Specialized data reduction software had to be developed to assimilate these data into a database that could be manipulated readily. Additional software was developed to generate summary performance measures and statistics. Even more challenging is the need to integrate the data sets so an analyst can move easily up and down the hierarchy exploring the data, developing an understanding of how lower level performance drives higher level outcomes, and developing an explanation of the results. It is the explanation of results that can be particularly challenging. This process extends beyond the manipulation of data output by the simulation. It requires that detailed knowledge of the mission environment and mission scenario, characteristics of the system being tested, and the operation of the human performance model be available during data analysis and that it is possible to access portions of these data to be able to answer questions as they arise.

Within the Case Study, the activity described above was accomplished through custom developed software or by more manual processing using simple tools such as spreadsheets or documents. The labor intensive nature of this method exhausted resources programmed for the data analysis rather quickly -- limiting the range of issues and questions that could be explored. The lesson learned is that CART testbeds generate a tremendous amount of data and information about a mission environment, but extracting that information and exploiting it to the maximum

extent possible can be difficult. Attention needs to be directed to developing an integrated set of tools that make the data reduction and analysis process more efficient.

This page left blank intentionally

REFERENCES

Barker, H.R. and Barker, B.M. (1984). *Multivariate Analysis of Variance (MANOVA): A Practical Guide to Its Use in Scientific Decision Making*. University of Alabama Press, University, Alabama.

Bray, J.H. and Maxwell, S. E. (1990). *Multivariate Analysis of Variance*. Sage Publications, Newbury Park, CA.

Brett, B.E., Doyal, J.A., Malek, D. A., Martin, E. A. and Hoagland, D. G. (2000). *The Combat Automation Requirements Testbed (CART) Task 1 Final Report: Implementation Concepts and an Example*, U.S. Air Force Research Laboratory Technical Report AFRL-HE-WP-TR-2000-0009. Wright-Patterson Air Force Base OH, 45433-7022.

Defense Modeling and Simulation Office (1998). *High Level Architecture*, [A Primer on CD-ROM] DMSO, Alexandria, VA 22311-1705.

DOD 5000.59-P (1995). *Modeling and Simulation (M&S) Master Plan*, Defense Modeling and Simulation Office (DMSO), Alexandria, VA 22311-1705.

Johnson, John (1958). *Analysis of Image Forming Systems*, Image Intensifier Symposium, Warfare Vision Branch, Electrical Engineering Dept., U.S. Army Engineer Research and Development Laboratories, Ft Belvoir, VA.

Martin, E.A., Brett, B. E. and Hoagland, D. G. (1999). *Tools for Including Realistic Representations of Operator Performance in DOD Constructive Simulations*, AIAA Modeling and Simulation Technologies Conference and Exhibit, Portland, OR. Paper Number AIAA-99-4027.

Neter, J. and Wasserman, W. (1974). *Applied Linear Statistical Models*, Richard D. Irwin, Inc., Homewood, IL.

Office of Aerospace Studies (2000). *AoA Handbook: A Guide for Performing an Analysis of Alternatives (AoA)*, Air Force Materiel Command (AFMC) OAS/DR, Kirtland AFB, NM.

Rasmussen, J.A., Pejtersen, M. and Goodstein, L. P. (1994). *Cognitive Systems Engineering*, John Wiley & Sons, New York.

Simulation Interoperability Standards Organization (1999). *Guidance, Rationale, and Interoperability Modalities for the Real-time Platform Reference Federation Object Model*, SISO-STD-001-1999, Orlando, FL.

SPSS for Windows (1999). SPSS, Inc., Chicago, IL.

This page left blank intentionally

ACRONYMS

A/G	Air to Ground
AIMS	Advanced Information Management System
AFF	Agile FOM Framework
AMS	Aircraft Management System
AoA	Analysis of Alternatives
ARL	Army Research Laboratory
CART	Combat Automation Requirements Testbed
C4ISR	Command, Control, Computer, Communications, Intelligence, Surveillance, and Reconnaissance
df	Degrees of Freedom
DMS	Display Management Switch
DMSO	Defense Modeling and Simulation Office
DOD	Department of Defense
DOI	Display of Interest
DTO	Defense Technology Objective
EFI	Electronic Flight Instrument
EOB	Electronic Order of Battle
ESAMS	Enhanced Surface-to-Air Missile Simulation
ESM	Electronic Support Measures
FMS	Flight Management Switch
FOM	Federation Object Model
FOV	Field of View
FRED	Fighter Requirements Evaluation Demonstrator
GCS	Generic Composite Scenario
GMTI	Ground Moving Target Indication
HDD	Head-Down Display
HITL	Human-in-the-Loop
HLA	High-Level Architecture
HOTAS	Hands-On Throttle and Stick
HPM	Human Performance Model
HRED	Human Research and Engineering Directorate
HUD	Head-Up Display
IFF	Identification Friend or Foe
IMPRINT	Improved Performance Research Integration Tool
IRST	Infrared Search and Track
JIMM	Joint Integrated Mission Model

JOBOB	Joint On-Board/Off-Board
JSF	Joint Strike Fighter
LAR	Launch Acceptance Region
M&S	Modeling and Simulation
MANOVA	Multivariate Analysis of Variance
MDU	Multi-Function Display Unit
MICS	Mission Interactive Combat Station
MOE	Measure of Effectiveness
MMD	Moving Map Display
MPD	Multi-Purpose Display
NM	Nautical Miles
OTW	Out-the-Window
RADGUNS	Radar Directed Gun Simulation
RPR	Real-time Platform Reference
RTI	Run-Time Infrastructure
RTMP	Real-Time Mission Planner
SA	Situation Awareness
SAM	Surface-to-Air Missile
SAR	Synthetic Aperture Radar
SBA	Simulation-Based Acquisition
SCRAMNET	Shared Common Random Access Memory Network
SGI	Silicon Graphics Inc.
SIMAF	Simulation and Analysis Facility
SIMAN	Simulation Management
SME	Subject Matter Expert
SOM	Simulation Object Model
SPSS	Statistical Package for the Social Sciences
SWEG	Synthetic Warfare Environment Generator
TBM	Theater Ballistic Missile
TCT	Time Critical Target
TIR	Targeting Infrared
TMS	Target Management Switch
TSD	Tactical Situation Display
UFC	Up-Front Control
UI	Unit Interdiction
VSWE	Virtual Strike Warfare Environment

APPENDIX A

CART CASE STUDY 1 ENVIRONMENT SELECTION PROCESS

This page left blank intentionally

INTRODUCTION

This document describes the process used to evaluate candidate topic areas for Case Study 1, and records the results of that process. The results of the process include a Case Study topic recommendation and some initial suggestions regarding how the integration might proceed. The sections in this document include the approach used to evaluate the topic area candidates, the description of the evaluation factors, other significant issues affecting the evaluation factors, and the results and recommendations from the evaluation.

THE EVALUATION APPROACH

This process was carried out in accordance with Task 3 of the CART statement of work (SOW). The SOW suggested some candidate topic areas, and defined the factors on which to evaluate the various topic areas. The approach focused on the Joint Strike Fighter (JSF) as a primary candidate for the first CART case study. However, the evaluation did not exclude other potential topic areas of interest to the Air Force Research Laboratory (AFRL). These other topic areas, listed in alphabetical order, included:

- The B-1B bomber defensive systems upgrade program (DSUP)
- C4ISR System of Systems (SoS)[6]
- Control Station 2010
- Information Operations/Sensors
- Uninhabited Combat Air Vehicles (UCAV)

The factors used to evaluate the candidate topic areas were:

- the types of human performance to be modeled
- the availability of existing system/environment models
- the cost/effort to create a constructive simulation

[6] Although the JSF program is considered a weapons acquisition program, the modeling and simulation effort currently being undertaken by JSF is aimed at understanding and defining this weapon system's role in a greater system – that of the Department of Defense's (DoD) Command, Control, Communications, Computing, Intelligence, Surveillance, and Reconnaissance (C4ISR) system. Therefore, this document refers to the JSF modeling and simulation topic area as the C4ISR System of Systems (SoS).

- availability of and cost/effort to condition Human-in-the-Loop (HITL) simulators
- availability of data required to generate performance measures
- program maturity/schedule fit with the CART program

Two other issues that were considered in the evaluation, but not specifically addressed in the SOW, were the CART team's domain expertise and the topic area's affiliation with the simulation-based acquisition (SBA) initiative.

Each candidate topic area was rated on each of the evaluation factors, and the results of these ratings were put into a matrix. The rows in the matrix corresponded to the topic areas, while the columns corresponded to the evaluation factors. In addition, the factors were weighted to reflect their relative importance in the evaluation. The weights were established through consideration of significant issues related to the evaluation factors that are explained later.

The candidate topic area that evaluated highest was recommended for Case Study 1. Finally, a development strategy was suggested for integrating the CART human performance model (HPM) into the selected constructive environment.

THE EVALUATION FACTORS

This section lists all the evaluation factors and describes each one in terms of this evaluation. They are presented in no particular order.

Types of human performance to be modeled

This factor referred to aspects of human behavior present in a system within the topic area to be modeled. The first aspect of human behavior was that it must have been complex enough to be of interest to CART. If the human behavior within the system were too simple it would not have been an effective demonstration of CART capabilities.

The second aspect of human behavior was that it must have been already defined sufficiently so that there existed at least one human behavior issue within the system that the target

program was interested in modeling. The CART capability of human performance modeling must have been of interest to the program.

Availability of existing system/environment models

This factor referred to the requirement for an existing constructive simulation testbed within the target program. In addition, the environment model (or mission model) needed to be able to generate mission-level events that could adequately exercise human-like interaction. Finally, the existing models needed to be accessible to the CART team.

Cost/effort to create constructive simulation

This factor was intended to sensitize the evaluation to the cost and effort required for creating or modifying an existing constructive simulation testbed of a candidate topic area. The goal was to reduce the risk to the program as much as possible. The more complete and appropriate the existing constructive environment was, the less cost, effort, and risk to CART Case Study 1 for building and integrating those models with CART software.

Availability of and cost/effort to condition HITL simulators

This factor was intended to sensitize the evaluation to the cost and effort required for creating or modifying an existing HITL simulation asset to simulate, as closely as possible, the scenarios played out in constructive simulation. The goal was to reduce the risk to the CART program as much as possible. The more complete and appropriate the existing HITL simulation environment, the less cost, effort, and risk to CART Case Study 1 for building and testing those simulators. In addition, this factor was intended to account for the expected availability of the target program's intended HITL simulation assets. While there was no expectation that the target program was to own its own HITL simulation assets, this factor captured the scheduling of simulation time on the HITL assets.

Availability of data required to generate performance measures

In order for the case study to fully demonstrate CART capabilities, the modeling and simulation environments had to allow for adequate data collection. In addition, data

collection and reduction was to have been relatively simple to implement with few technical or administrative challenges. In this case, an example of a technical challenge may have been the need to author a data collection engine from scratch. An example of an administrative challenge may have been that the data were too highly classified to be released to -- or used by -- the CART program.

Program maturity/schedule fit with CART

This factor referred to the requirement that the target program must have been mature enough to have already established a modeling and simulation program, and that modeling and simulation would be conducted within the time frame established by the CART case study schedule. In addition, the modeling and simulation activities were to have been at the appropriate stage of the program life cycle (i.e., the modeling and simulation efforts were to be aimed at establishing requirements at some level).

Team's domain expertise

Although this factor was not one specified by the statement of work, it was the intent of this evaluation to reduce risk to the CART program by choosing a case study topic area in which the team had at least some domain expertise.

Affiliation with Simulation-Based Acquisition (SBA) initiative

This factor, again not one specified by the SOW, was intended to establish affiliation with the SBA initiative in order to better market CART within the modeling and simulation community. It was decided that the target program should be seeking to define and/or extend SBA methods and tools, and that the target program should help advocate the use of CART within the SBA community.

SIGNIFICANT CASE STUDY ISSUES RELATED TO THE EVALUATION FACTORS

There were two major issues not included in the evaluation factors, but they affected the weighting of individual evaluation factors. The first was implications surrounding the

requirement for virtual (HITL) simulation needed to verify and validate the constructive HPM simulation. The second was perceived benefit of CART to the target program, and how this perception may impact the future of CART. These two issues are now addressed individually.

Case study requirement for virtual simulation needed to verify and validate constructive data

In order to demonstrate the validity of a HPM provided by CART, each case study requires that virtual simulation be conducted using the same scenarios played by the constructive environment. A technical challenge -- given this requirement -- is the need for controlling the differences between constructive and virtual worlds that could affect mission outcomes or mission performance. For the validation exercise to be the most effective, differences in mission outcomes need to be attributable to, as much as possible, the human operator in virtual simulation and the HPM in constructive simulation -- not to differences in the simulation environments. To accomplish this end, the differences between the constructive and virtual worlds need to be as few as possible and as completely described as possible. Therefore, the factors associated with the simulation environments were given the most weight in consideration of this extremely important issue.

Perceived benefit of CART to target program and target program advocacy

Another important issue considered in assigning evaluation factor weights was the expected benefit of CART to the target program to maximize the case study's demonstration of CART utility to the acquisition community. To accomplish this end, the target program should have had a demonstrated or stated need for the technology being offered by CART. In addition, it was desirable to choose a target program that understood and supported CART program goals and was willing to act as an advocate for these goals within the modeling and simulation community.

RESULTS AND RECOMMENDATIONS

Table A-1 shows the topic area evaluation matrix. The columns correspond to the evaluation factors while the rows correspond to the candidate topic areas. The factor category weights are shown with the column titles.

Table A-1. Case Study 1 Topic Area Evaluation Matrix

Topic Area (category weight)	Human Performance Complexity (4)	Current Modeling Infrastructure (5)	Constructive Sim Effort (3)	HITL Sim Effort (3)	Sim Data Availability (2)	Program Maturity/ Schedule Fit (3)	Team's Domain Expertise (2)	Affiliation with SBA (3)	Measure of Probable Success
C4ISR SoS	High (3)	Good (3)	Moderate (2)	Moderate (2)	High (3)	Good (3)	Good (3)	High (3)	Good (.91)
UCAV	High (3)	Poor (1)	High (1)	High (1)	Low (1)	Fair (2)	Good (3)	Moderate (2)	Poor (.59)
Control Station 2010	Undefined	Undefined	Undefined	Undefined	Undefined	Poor (1)	Fair (2)	Undefined	Poor (.09)
B-1 DSUP	Moderate (2)	Poor (1)	High (1)	High (1)	Low (1)	Poor (1)	Excellent (4)	Low (1)	Poor (.45)
Information Ops/ Sensors	Low (1)	Poor (1)	High (1)	High (1)	Low (1)	Poor (1)	Fair (2)	Moderate (2)	Poor (.41)
Ratings	3 – High 2 – Moderate 1 – Low	3 – Good 2 – Fair 1 – Poor	3 – Low 2 – Moderate 1 – High	3 – Low 2 – Moderate 1 – High	3 – High 2 – Moderate 1 – Low	3 – Good 2 – Fair 1 – Poor	3 – Good 2 – Fair 1 – Poor	3 – High 2 – Moderate 1 – Low	

The ratings were developed from discussions with topic area program personnel and from research into the candidate programs. The score called Measure of Probable Success is expressed as a fraction of the possible total points calculated by multiplying each topic area rating by its category weight and summing that result for each topic area. A 1.00 represented 100% probability of case study success, where success was considered as meeting all the requirements for conducting a case study under the terms of the contract and demonstrating that CART technology was beneficial to the modeling and simulation community.

Topic Area Evaluation Matrix Results

The C4ISR SoS scored very well primarily due to the close coupling of its constructive and virtual modeling and simulation environments, but the other factors were also rated highly. The UCAV topic area scored lower primarily because this program is extremely early in its life cycle, and because the program currently has a strong contractor influence. UCAV will likely be a top candidate during the evaluation of topic areas for Case Study 2. The Control

Station 2010 topic area, an advanced command and control workstation concept, scored lowest primarily due to the current lack of a clear program definition. It is expected that this topic area will also be revisited during the Case Study 2 evaluation. The B-1 DSUP topic area evaluated lower than some of the other topic areas primarily due to the lack of a stated need by the B-1 program for human performance modeling, especially in the DSUP program. Since the Information Operations/ Sensors topic area was quite broad, it was expected that the chance of finding a suitable target program from this domain was high. However, the primary reason for the lack of high evaluation score was the current lack of interest in the CART technology by the relevant organizations within Electronic Systems Center (ESC) and Space and Missile Center (SMC), and the lack of any well-defined constructive modeling and simulation testbeds. This topic area will also be revisited during the Case Study 2 evaluation.

Clearly, the C4ISR SoS evaluated very highly in the matrix. It was discovered that the C4ISR SoS Virtual Strike Warfare Environment (VSWE), which is the modeling and simulation testbed for JSF, provided the best possible solution to the challenge. It offered a seamless transition between the virtual and constructive environments -- consequently eliminating the problem of controlling for differences in simulation results unrelated to the HPM vs. the human operator issue. CART also seems to provide a clear benefit to the JSF program in terms of filling a stated modeling need, as the JSF program has already emerged as a strong CART advocate. In addition, the JSF modeling and simulation program has a very high visibility within the Air Force and across the DoD. Finally, the JSF modeling and simulation program is currently considered a leader in the SBA community, and the CART program should benefit greatly from this affiliation. No other topic areas could match these benefits.

Recommendations

The recommendation from this analysis is that the first CART case study topic area should be the JSF C4ISR SoS. However, all other topic areas considered here should be considered for evaluation again during the search for a topic area for Case Study 2. The next section of this report proposes a framework for integrating a CART HPM into the existing C4ISR SoS VSWE modeling and simulation testbed.

CART INTEGRATION FOR CASE STUDY 1

Virtual Strike Warfare Environment (VSWE)

The current modeling and simulation testbed in place for JSF is the VSWE. Figure A-1 shows the basic models and their interrelationships. The different modules of the simulation environment are called the FRED, the SWEG, and the SWEDAT.

FRED

FRED stands for the *Fighter Requirements Evaluation Demonstrator* and provides the virtual cockpit management software, controls the avionics and aero models, and manages the controls, displays, and out-the-window scene generation software.

SWEG

SWEG stands for *Simulated Warfare Environment Generator* and is the mission-level modeling environment. It is an event-stepped, data-driven modeling environment that provides the terrain, targets, threats, and all C4ISR data and events.

SWEDAT

SWEDAT stands for *SWEG data interface* and is an area in shared memory that allows assets external to SWEG to dynamically interact with SWEG. The SWEDAT typically resides in a SCRAMNET shared memory interface.

CART HPM Integration into the VSWE

A proposed architecture for integrating a CART HPM into the current VSWE environment is shown in Figure A-2. The current VSWE environment is shown at the left, the CART HPM is shown at the right, and a proposed High-Level Architecture (HLA) Run-Time Infrastructure (RTI) is shown at center, linking the two environments.

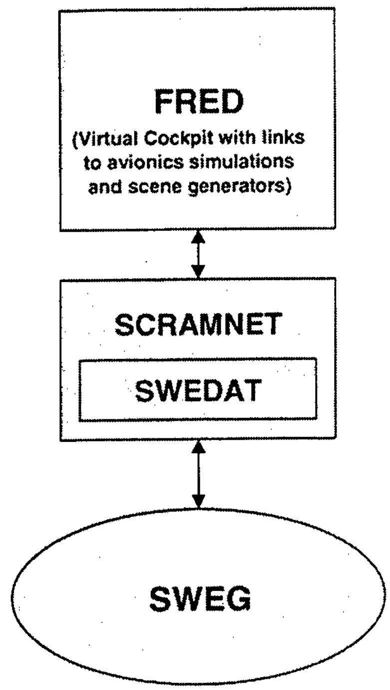


Figure A-1. A Simple Look at the Current VSWE Environment

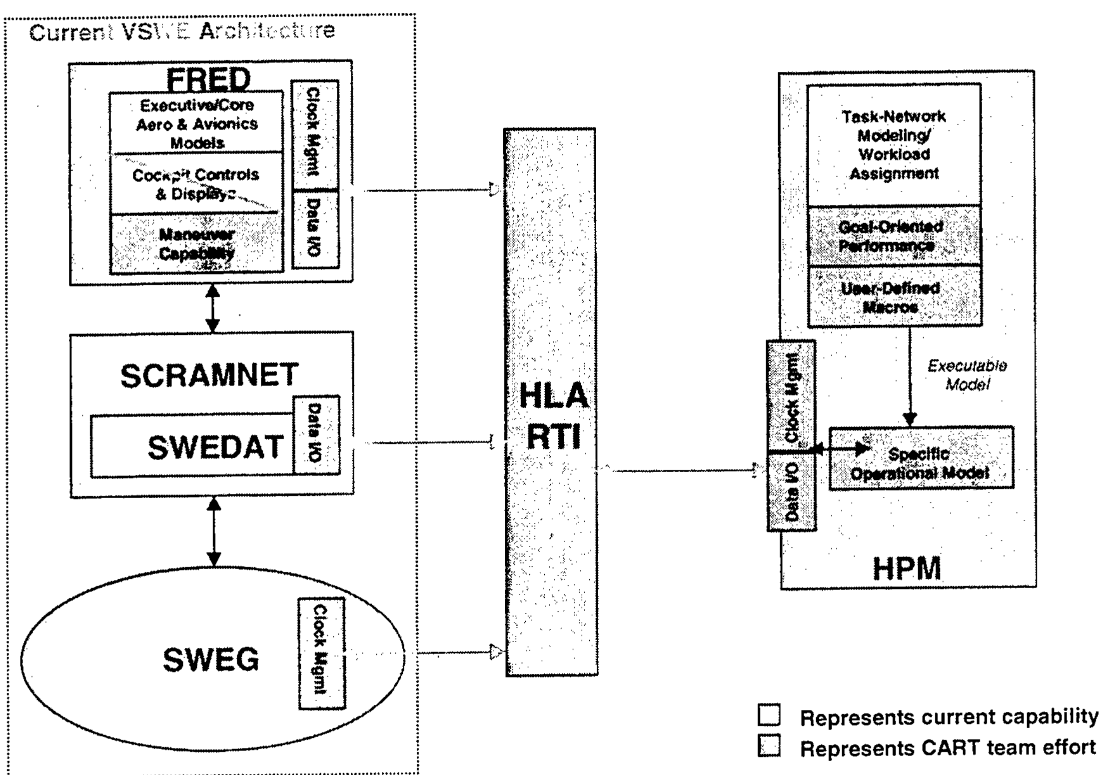


Figure A-2. CART HPM Architecture Integration with the VSWE

This page left blank intentionally

APPENDIX B

TASK NETWORK DIAGRAMS FOR THE CASE STUDY 1 HUMAN PERFORMANCE MODEL

This page left blank intentionally

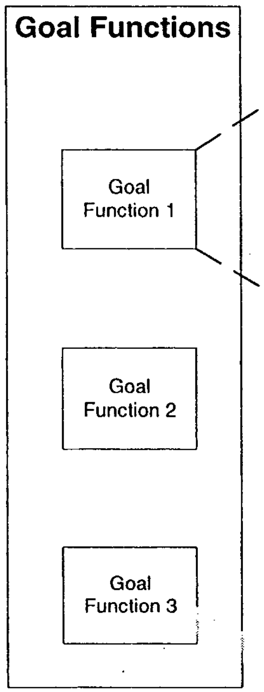
Overview of Task Network Symbology

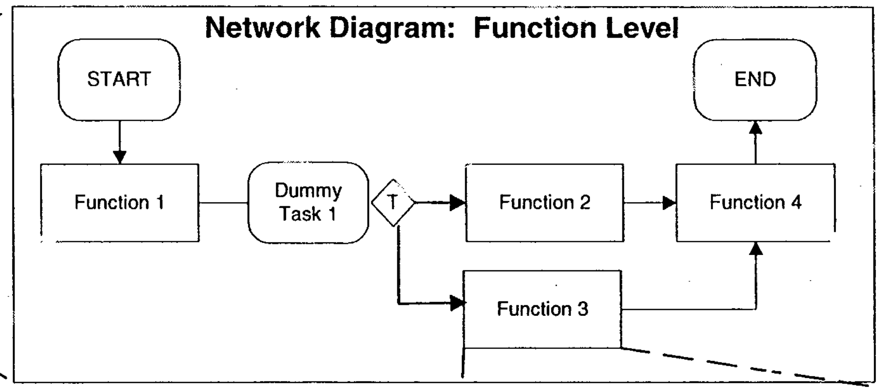
This appendix contains all of the task networks that comprised the Case Study 1 HPM. Some of the terms and symbology used in these diagrams are described below.

Dummy Task: A task that has no associated time or workload. It is often used to allow the user more modeling flexibility.

Start and End Tasks: A form of dummy task that specifies the beginning and ending of a given task network. These tasks have no effects or release conditions.

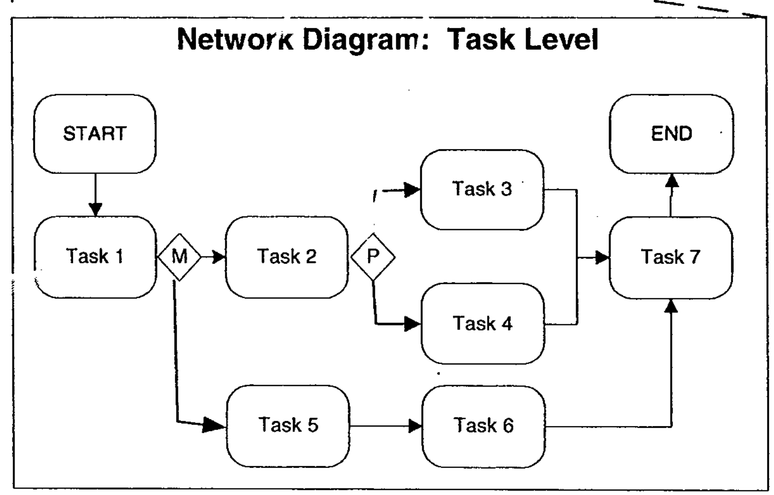
Tactical Decision: Allows the user to define task routing based upon the current value of a given expression.





Multiple Decision: Allows the user to initiate multiple, simultaneous events within a model.

Probabilistic Decision: Allows the user to specify a probability of occurence for various task. At runtime, a random number generator determines which task will execute.



Function: A means of organizing lower level tasks or functions into a meaningful grouping. A function can be decomposed into a set of functions or tasks. In the network diagrams, a function is represented by a rectangle.

Goal Function: The highest level of organization within a model. Goal functions are not directly networked in diagrams. Rather, the user defines conditions that trigger the goal function and also prioritizes the goals.

Task: The basic building block of a model. A task is defined by timing and accuracy information, workload information, release conditions, and beginning and ending effects. In the network diagrams, a task is represented by a rectangle with rounded corners.

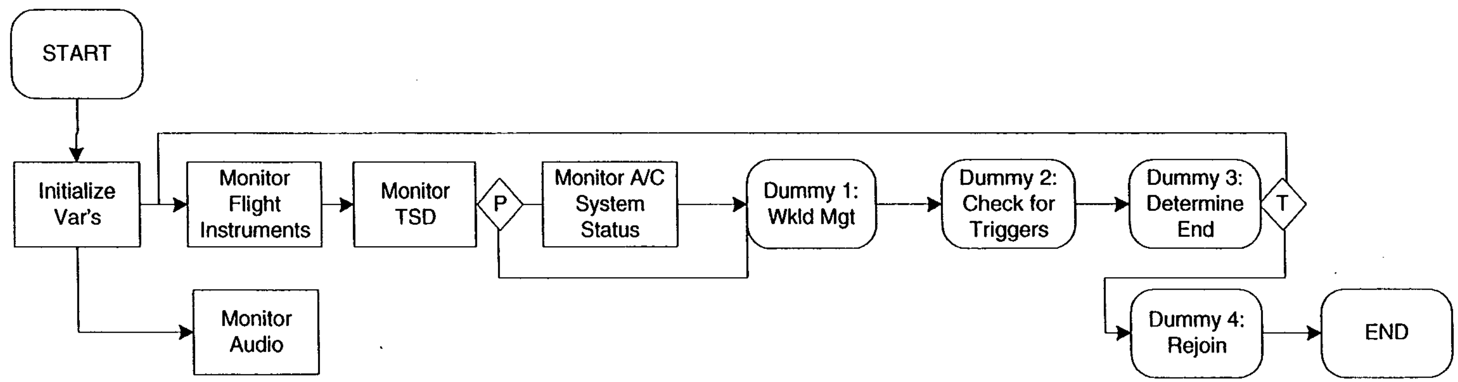


Figure B-1. "Mission"-Level Function Network (represents the *Control Aircraft/ Maintain SA* Goal Function)

Description: This diagram represents the "mission"-level network. It includes a function to initialize the variables and then it begins a loop that represents the pilot's typical instrument scan and listening function. A probabilistic decision determines the frequency with which the *Monitor A/C System Status* Task fires, as system status is typically checked less frequently than the primary flight instruments and the tactical situation. Dummy tasks are included to handle workload timesharing with other tasks, to check trigger conditions, and to end the trial.

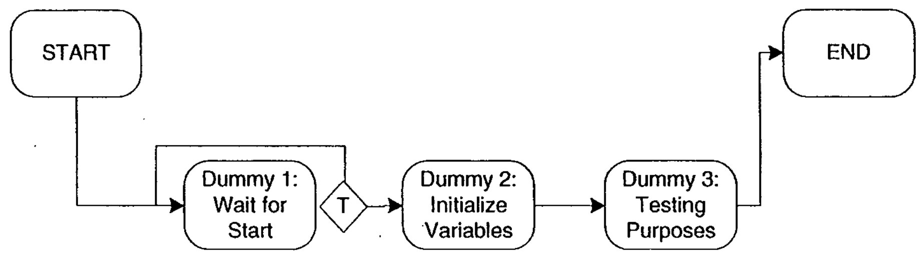


Figure B-2. Task Network for *Initialize Var's* Function Within Mission Level Function Network

Description: The *Initialize Var's* function consists of dummy tasks that execute prior to the start of the trial. A repeating task waits for a start message from the FRED. Once this is received, all model variables are initialized. After initialization, a test task was inserted to debug timing problems during integration. For data collection, this task was given a one second duration to ensure all FRED data was available prior to trial startup. Effects in this task were disabled.

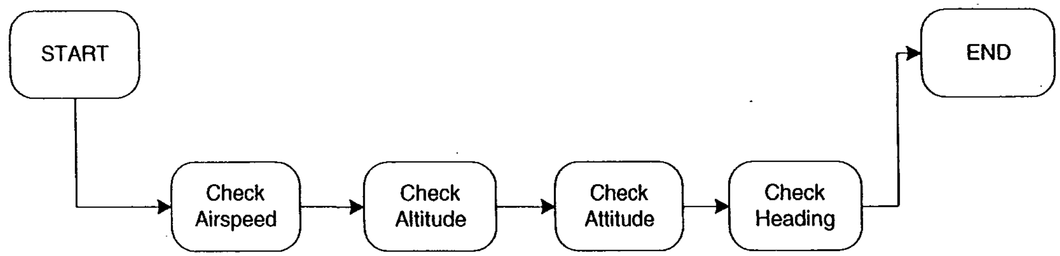


Figure B-3. Task Network for *Monitor Flight Instruments* Function Within Mission Level Function Network

Description: During the *Monitor Flight Instruments* task, airspeed, altitude, attitude and heading are checked. Internal variables for each of these parameters are updated to reflect the current values supplied by the FRED at the time of task execution. These internal variables reflect the pilot's "perceived" value of the parameters.

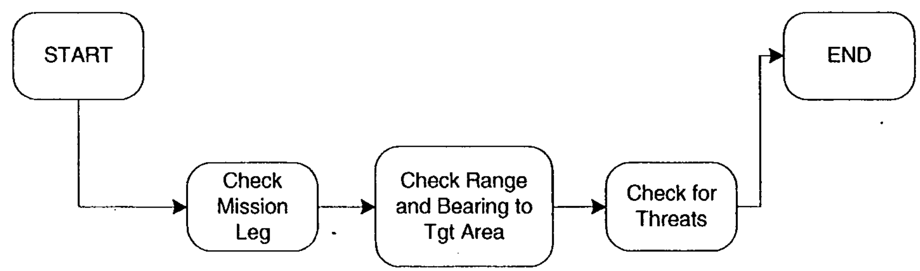


Figure B-4. Task Network for *Monitor TSD* Function Within Mission Level Function Network

Description: This function represents the pilot examining the tactical situation display. In the *Check Mission Leg* task, the perceived aircraft position is updated and a macro is used to determine whether critical waypoints have been sequenced. Next, the perceived range and bearing to the target area are updated. These data are used to determine whether the target area is within range of the sensors, and if so, to trigger the *Acquisition* function. The *Check for Threats* task represents the pilot's checking known and pop-up threat sites and monitoring for launch symbols.

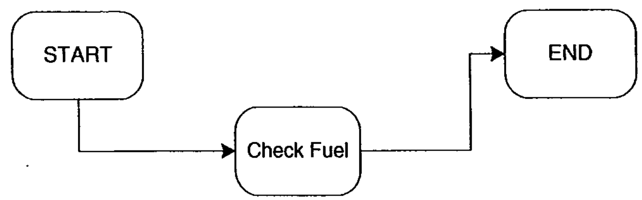


Figure B-5. Task Network for *Monitor A/C System Status* Function Within Mission Level Function Network

Description: In the VSWE scenario, there were no system failures. Thus, the only system status information to be updated was the fuel state. In the *Check Fuel task*, the perceived fuel amount was set to the actual amount.

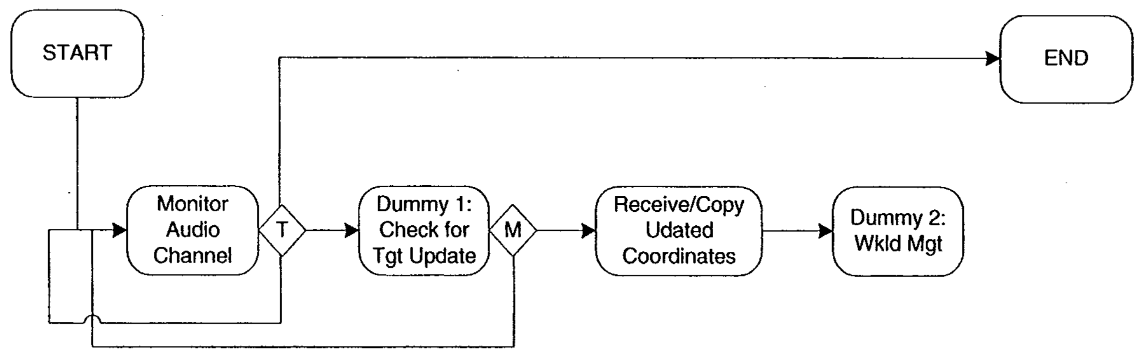


Figure B-6. Task Network for *Monitor Audio* Function Within Mission Level Function Network

Description: In the *Monitor Audio* function, a *Monitor Audio Channel* task repeats continuously. This task checks for threat and planner tones as well as radio communications of a target update. If a target update message is received, a multiple pathway is initiated such that the pilot continues to monitor for threat/planner tones while at the same time recording the target coordinates received in the message. A dummy task is also used to employ the workload management scheme while the target update message is being recorded.

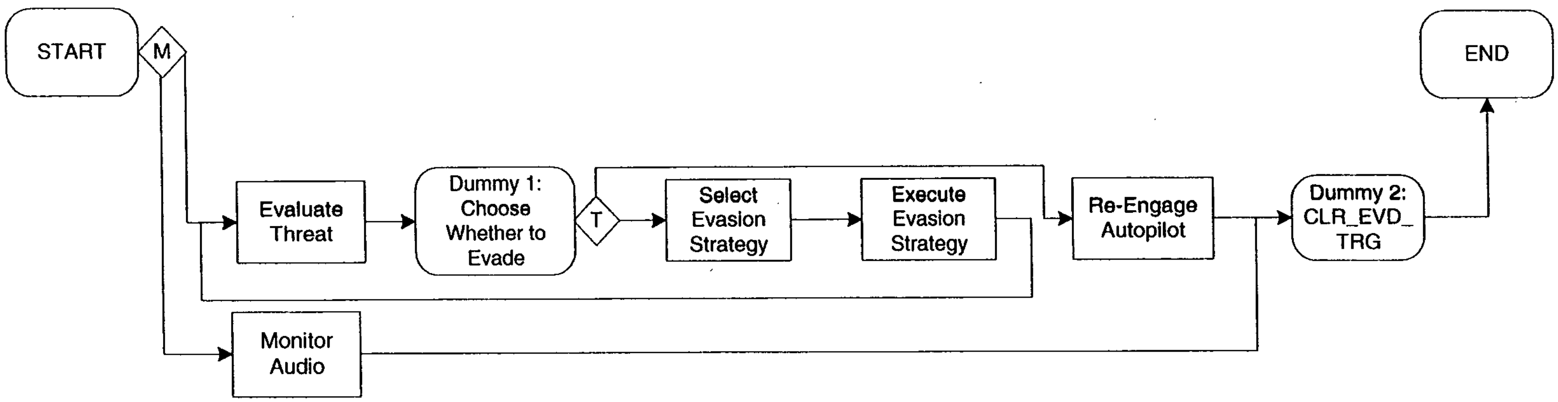


Figure B-7. ***Evade*** **Goal Function Network**

Description: This diagram shows the functions and tasks that support the *Evade* goal function. In this function, the threat is first evaluated. If it is considered to be lethal given its range to the aircraft, an evasion strategy is selected and employed and the threat is re-evaluated. If it is still active, the process is repeated. Once the threat is no longer present (i.e., the missile has missed and the threat radar is no longer in launch mode), the autopilot is re-engaged. During the *Evade* function, the audio channel is also monitored for additional threat tones.

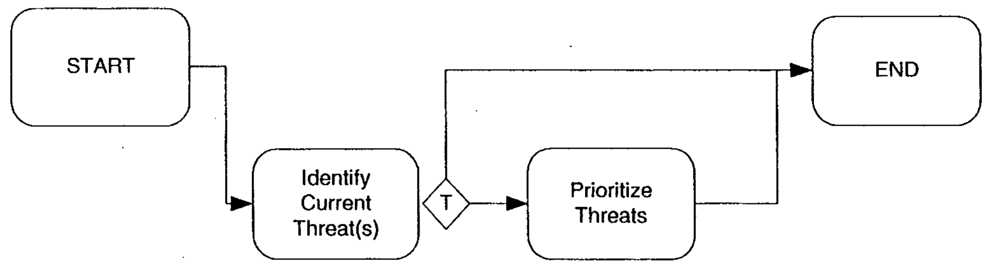


Figure B-8. Task Network for *Evaluate Threat* Function Within *Evade* Goal Function

Description: The *Evaluate Threats* function begins with a task in which the current threat or threats are identified. In this task, the type and range of any active missiles are determined in order to make a judgment whether they are "lethal" to the aircraft. If more than one missile is active and considered lethal, a threat prioritization task executes in which the threats are prioritized based on their estimated time to intercept. All subsequent evasive actions are intended to defeat the highest priority threat.

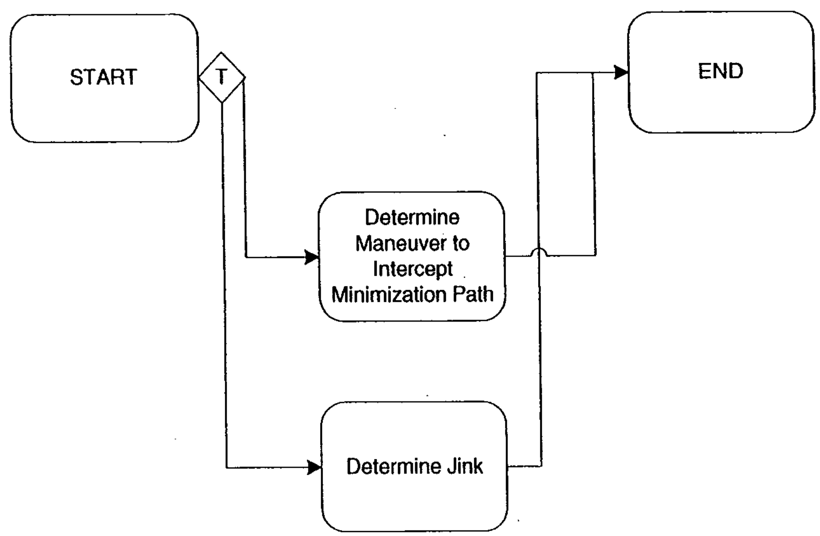


Figure B-9. Task Network for *Select Evasion Strategy* Function Within *Evade* Goal Function

Description: *The Select Evasion Strategy* function consists of two tasks. For any given threat, on the first pass through the *Evade* function loop, the *Determine Maneuver to Intercept Minimization Path* task is executed. In this task, a desired aircraft heading for evading the missile is calculated. If the missile is still active after the first attempt at evasion, subsequent passes through this function will cause the *Determine Jink* task to execute. In this task, a new desired heading will be calculated, representing a pilot's desire to jink to one side or the other of the evasive heading.

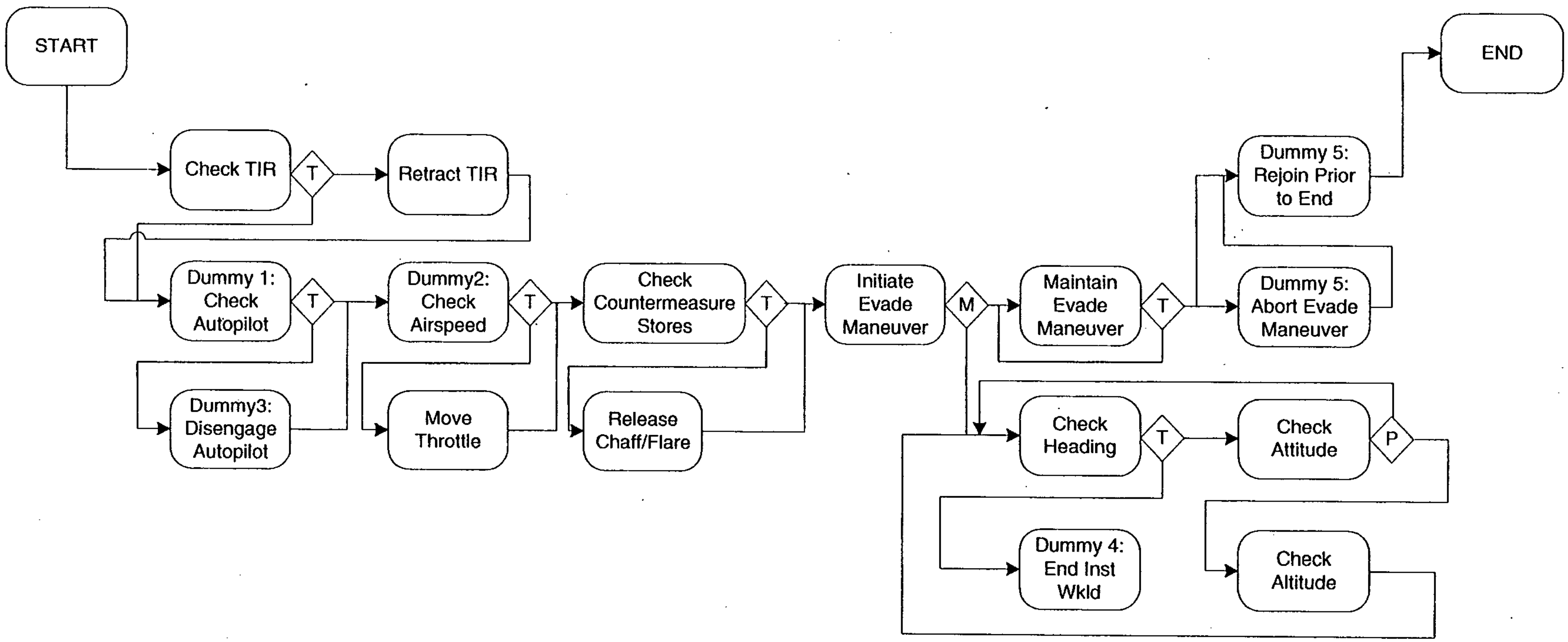


Figure B-10. Task Network for *Execute Evasion Strategy* Function Within *Evade* Goal Function

Description: The *Execute Evasion Strategy* function consists of the tasks necessary to perform manual aircraft maneuvering in an effort to evade a launched missile. It consists of tasks to stow the TIR if deployed, go into afterburner, release countermeasures, and perform a manual turn to the evasive heading. Throughout the duration of the evasive turn, heading, attitude and altitude are monitored.

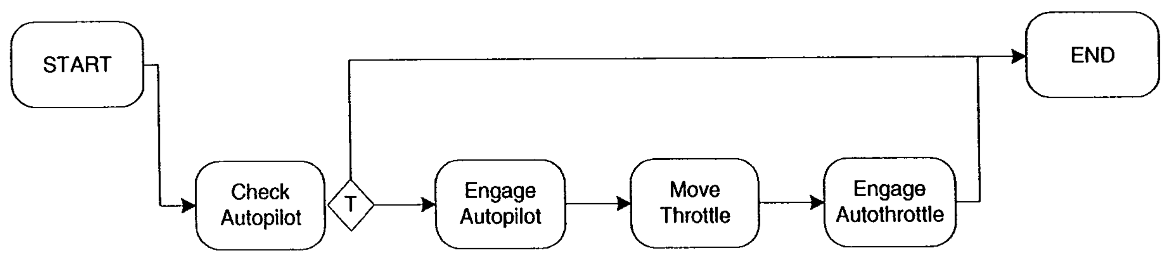


Figure B-11. Task Network for *Re-Engage Autopilot* Function Within *Evade* Goal Function

Description: Once the missile threat is no longer active, the *Re-Engage Autopilot* function is performed. It involves engaging the autopilot (if necessary) setting the throttle back to its cruise position, and re-engaging the autothrottle.

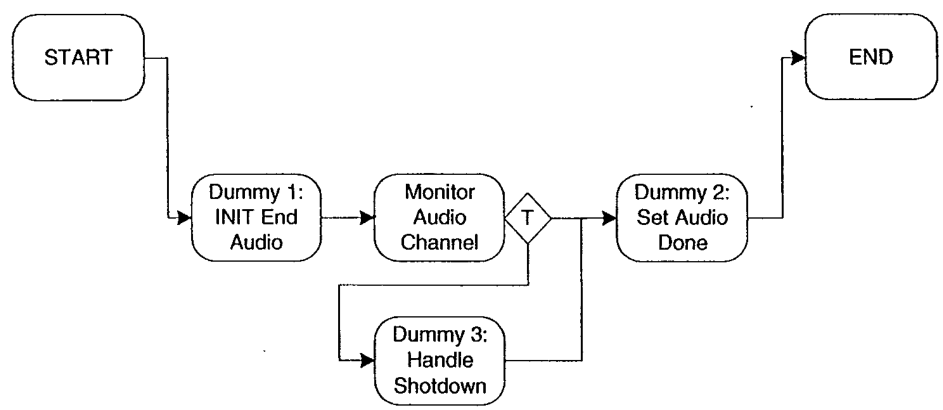


Figure B-12. Task Network for *Monitor Audio* Function Within *Evade* Goal Function

Description: During the course of the *Evade* goal function, the *Monitor Audio* function is active. This includes a *Monitor Audio* Task that continues to monitor for new threat launch tones and a set of dummy tasks to control the start/end of the audio task and initiate model termination if the aircraft is shot down (this task's effects were disabled during data collection).

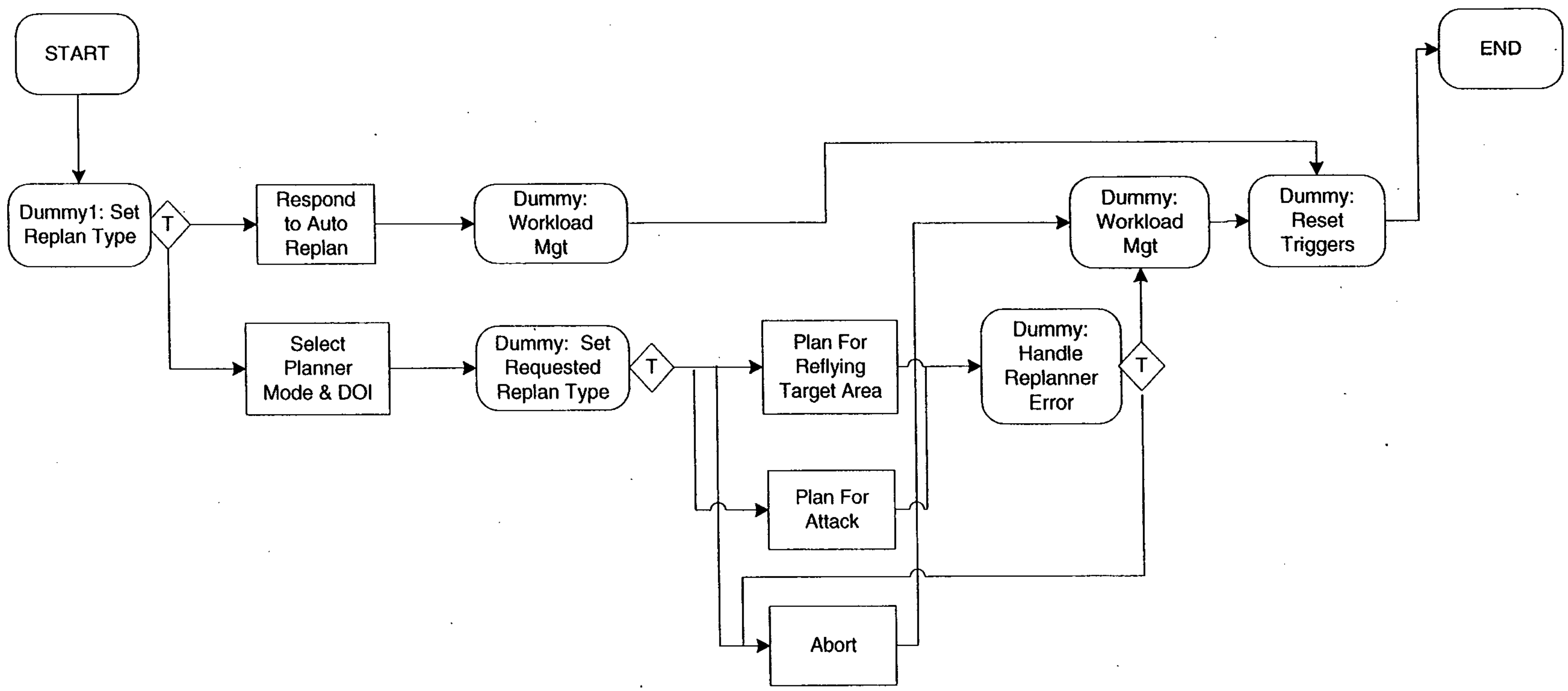


Figure B-13. ***Navigate*** **Goal Function Network**

Description: The *Navigate* Goal Function performs both acceptance of auto re-plans and generation of manual re-plans. In the case of manual re-plans, the planner mode is selected and then the plan is requested. Types of manual re-plans include planning to refly the target area, planning to attack an identified target and planning to abort.

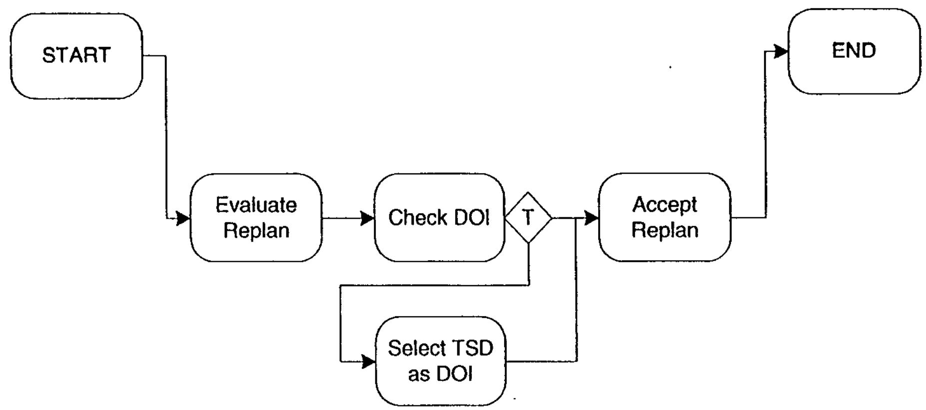


Figure B-14. Task Network for *Respond to Auto Re-plan* Function Within *Navigate* Goal Function

Description: This function executes when an auto re-plan is presented by the mission planner. The re-plan is evaluated, the display of interest is set to the TSD if necessary, and the re-plan is accepted. For the purposes of Case Study 1, all generated re-plans were accepted.

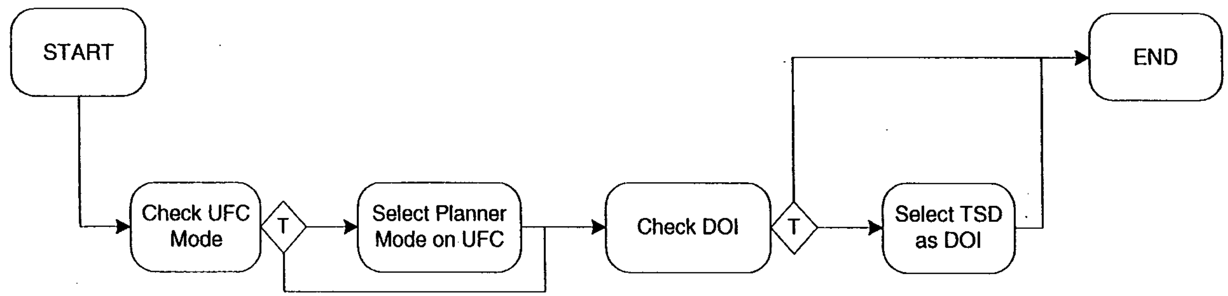


Figure B-15. Task Network for *Select Planner Mode and DOI* Function Within *Navigate* Goal Function

Description: The Select Planner Mode and DOI function is executed prior to requesting a manual re-plan. Tactical decisions check the UFC mode and display of interest and execute tasks to set them if necessary.

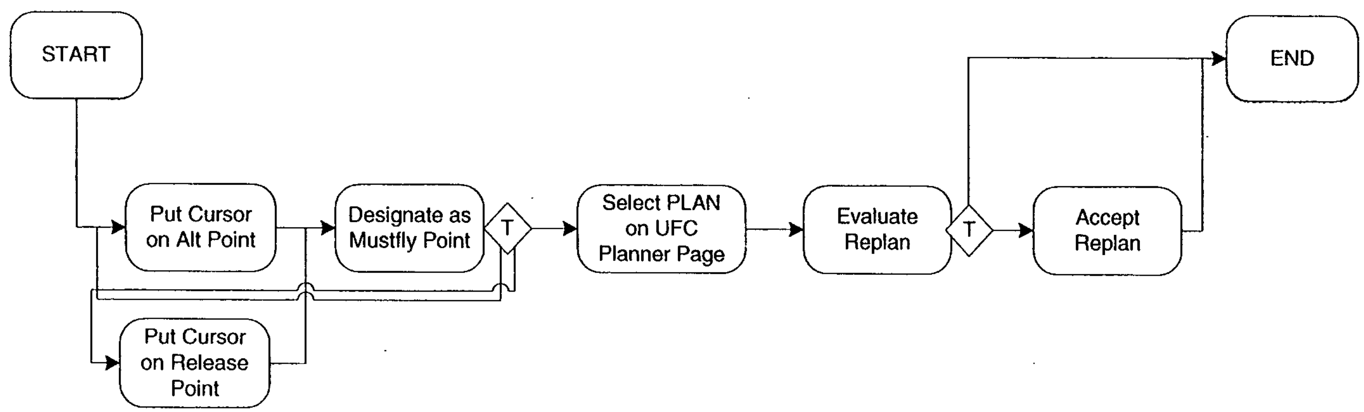


Figure B-16. Task Network for *Plan For Reflying Target Area* Function Within *Navigate* Goal Function

Description: This manual re-planning function is executed if the original weapon release point is sequenced prior to the target being identified. Two alternate waypoints and the original weapon release point are designated on the TSD to define the mustfly points for reflying the acquisition leg. The plan is then requested, evaluated, and then accepted once it is returned by the planner.

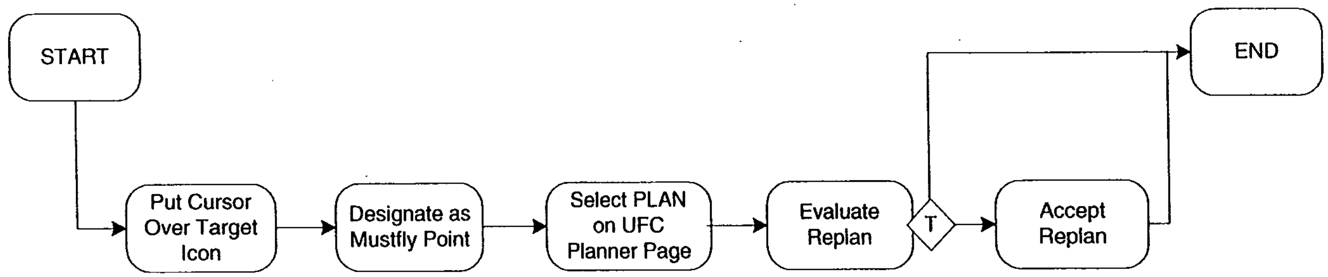


Figure B-17. Task Network for *Plan For Attack* Function Within *Navigate* Goal Function

Description: The *Plan for Attack* function is performed if the identified target is not directly on the aircraft's immediate route. The cursor is slewed to the target and the target is designated to create a mustfly point (at altitude). The manual re-plan is then requested, evaluated and accepted.

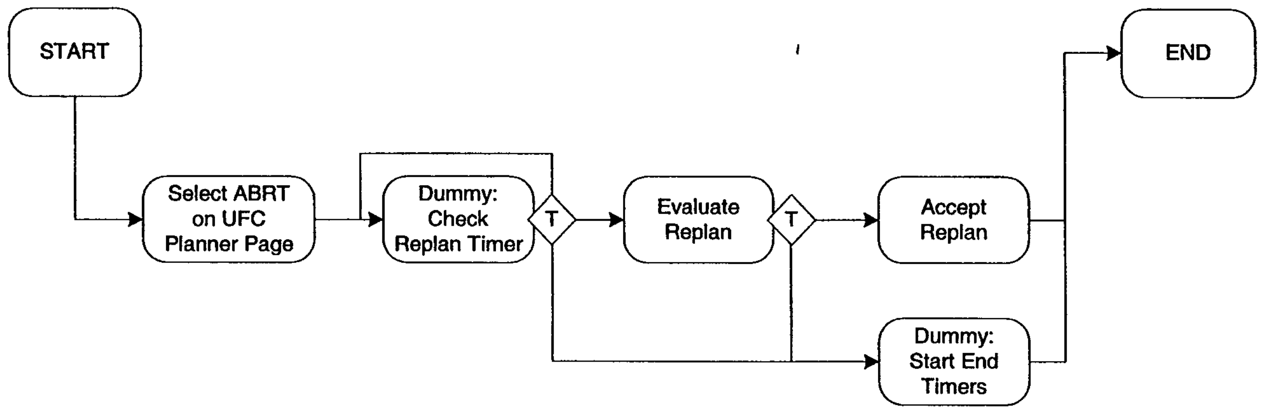


Figure B-18. Task Network for *Abort* Function Within *Navigate* Goal Function

Description: The *Abort* function occurs after the target has been successfully attacked or after flying two passes over the target area without identifying the target. In either case, the abort is requested, evaluated and then accepted. Dummy tasks are used to check for re-plan availability and to initiate a timer used to end the trial after an abort is requested.

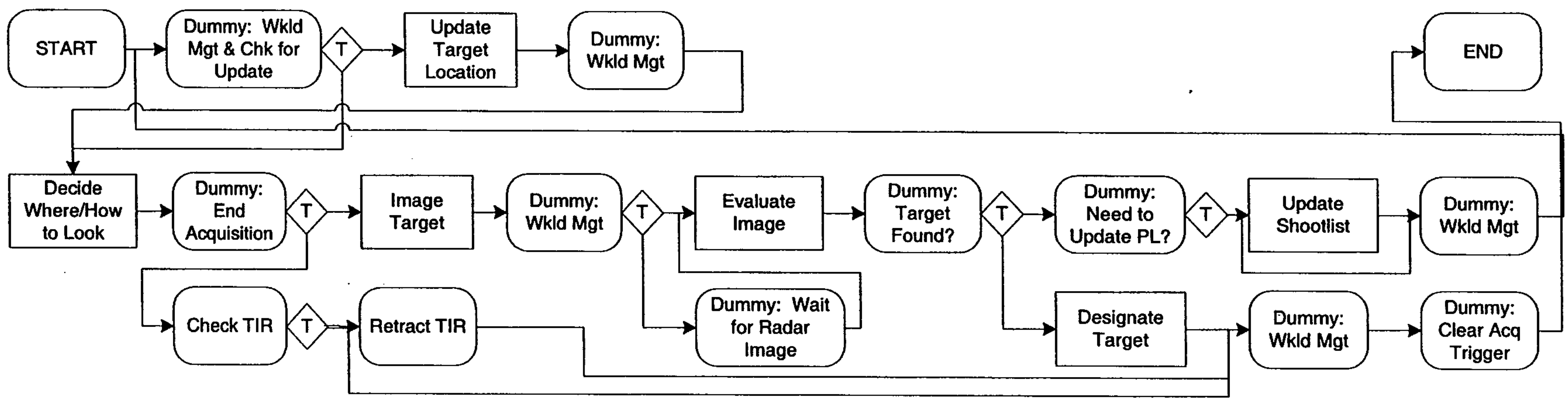


Figure B-19. ***Acquire*** **Goal Function Network**

Description: The *Acquire* goal function consists of an acquisition loop that is repeated each time that a new image is required. The loop begins with a decision about where to look and what sensor to use. The sensor is then employed to image the target, and the resulting image is evaluated. If the target is identified, the target is then designated for attack. If the target is not identified, the shootlist is modified based on the results of the image evaluation, and the acquisition loop is repeated. In addition, the *Acquire* goal function includes pilot actions to update the target location in the mission planner once the target update is received. A number of dummy tasks are used to check for the update, manage workload, and to clear trigger conditions.

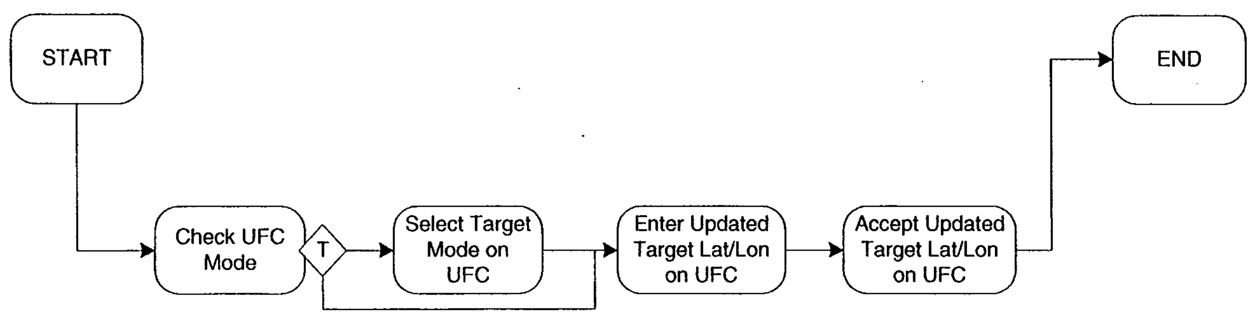


Figure B-20. Task Network for *Update Tgt Location* Function Within *Acquire* Goal Function

Description: This function represents pilot activity for entering the updated target coordinates into the planner. The target mode of the planner is set, if necessary, and the latitude and longitude of the updated point is keyed into the system. Once entered, the updated target position is accepted.

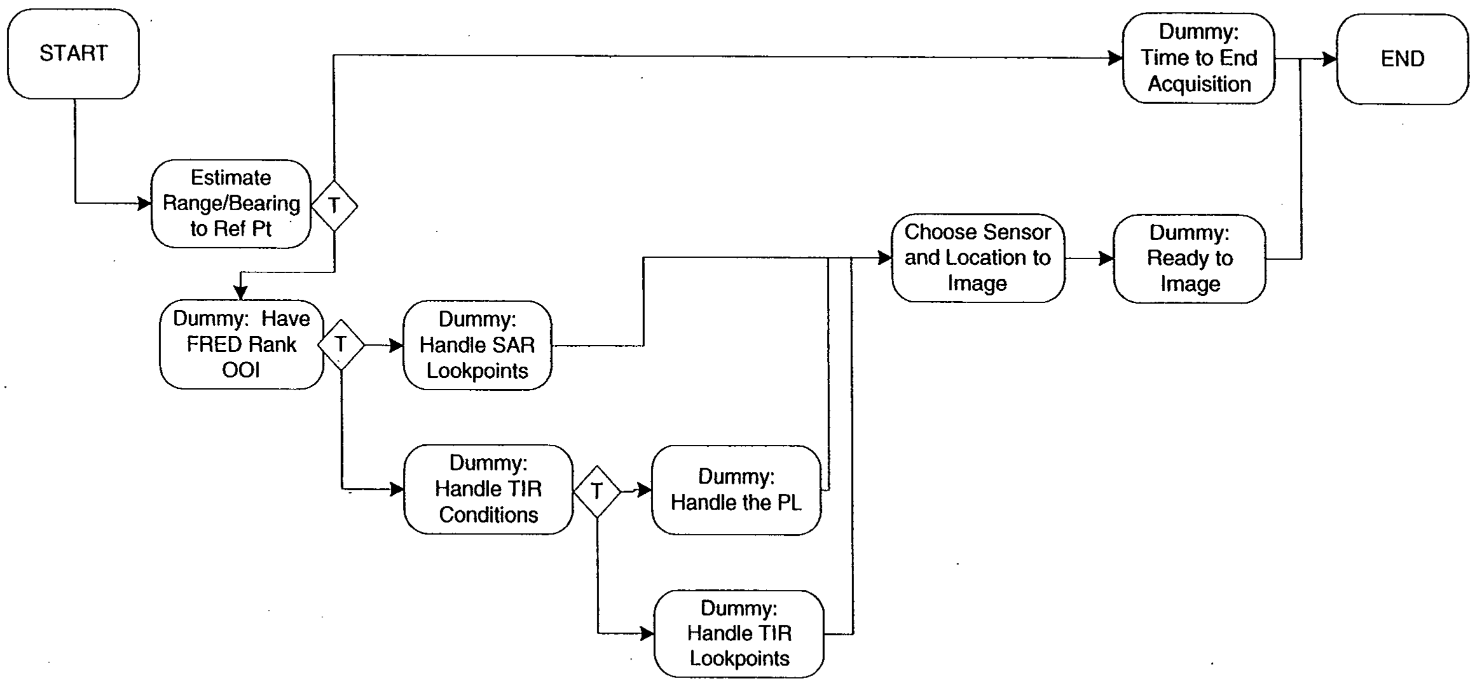


Figure B-21. Task Network for *Decide Where/How to Look* Function Within *Acquire* Goal Function

Description: This function represents the pilot's decision regarding where to look on the ground and which sensor to use. It begins with a task to estimate the distance to the target area. Next, a number of dummy tasks are used to step through a decision process. These include ranking the objects of interest based on their range to the reference point (i.e., the best estimate of the target's location) and whether they are moving, considering alternate lookpoints for both the SAR and TIR, and choosing the next appropriate object off of the priority list. Next, the *Choose Sensor and Location to Image* operator task assigns time and workload to the decision making task. Dummy tasks are also included to end the *Acquire* function or to enable imaging the target as appropriate.

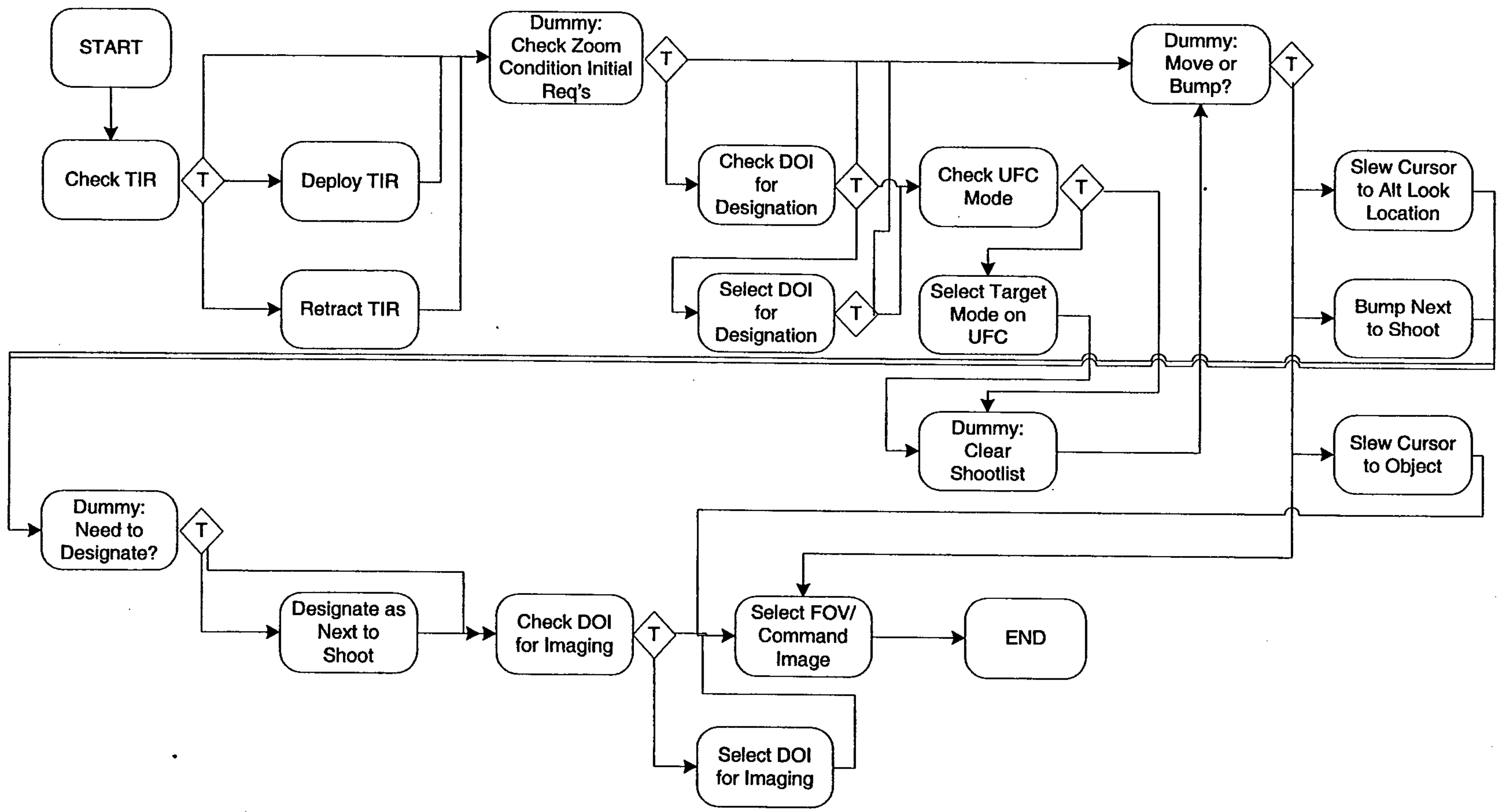


Figure B-22. Task Network for *Image Target* Function Within *Acquire* Goal Function

Description: Here, the tasks required for sensor employment are represented They include stowing/deploying the TIR as necessary, selecting the radar, TSD, or TIR as the display of interest, slewing or bumping the cursor as necessary and commanding the image.

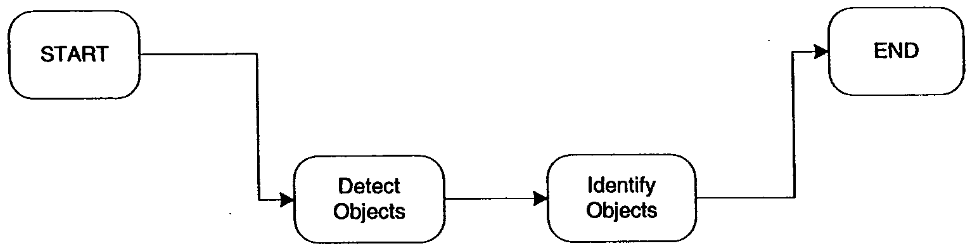


Figure B-23. Task Network for *Evaluate Image* Function Within *Acquire* Goal Function

Description: In this function, the target detection and identification tasks are represented. Both tasks call macros that use Johnson's Criteria[7] to calculate whether any object in the sensor field of view would be detected and identified based upon its appearance on the cockpit display. If an object is detected, identified, and found to be the same object type as the target of interest (i.e., a TBM) then it is designated for attack (see following function diagram) and the *Acquire* function or "goal" function ends.

Johnson, John (1958). Analysis of Image Forming Systems, Image Intensifier Symposium, Warfare Vision Branch, Electrical Engineering Dept. U.S. Army Engineer Research and Development Laboratories, Ft Belvoir VA.

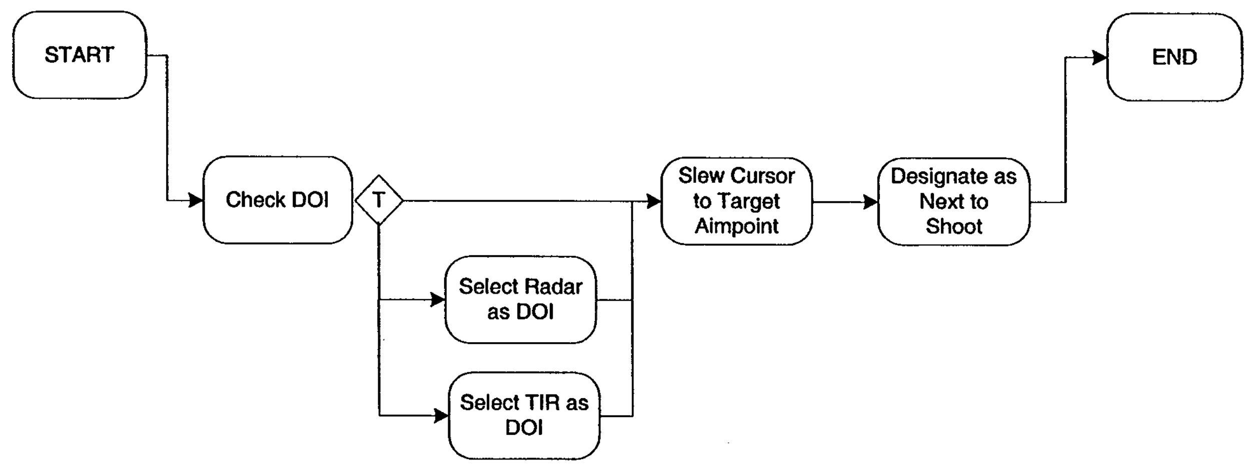


Figure B-24. Task Network for *Designate Target* Function Within *Acquire* Goal Function

Description: Once the target is identified, this function is used to designate that target as "next to shoot". The display of interest is selected as appropriate, the cursor is slewed to the target, and the target is designated.

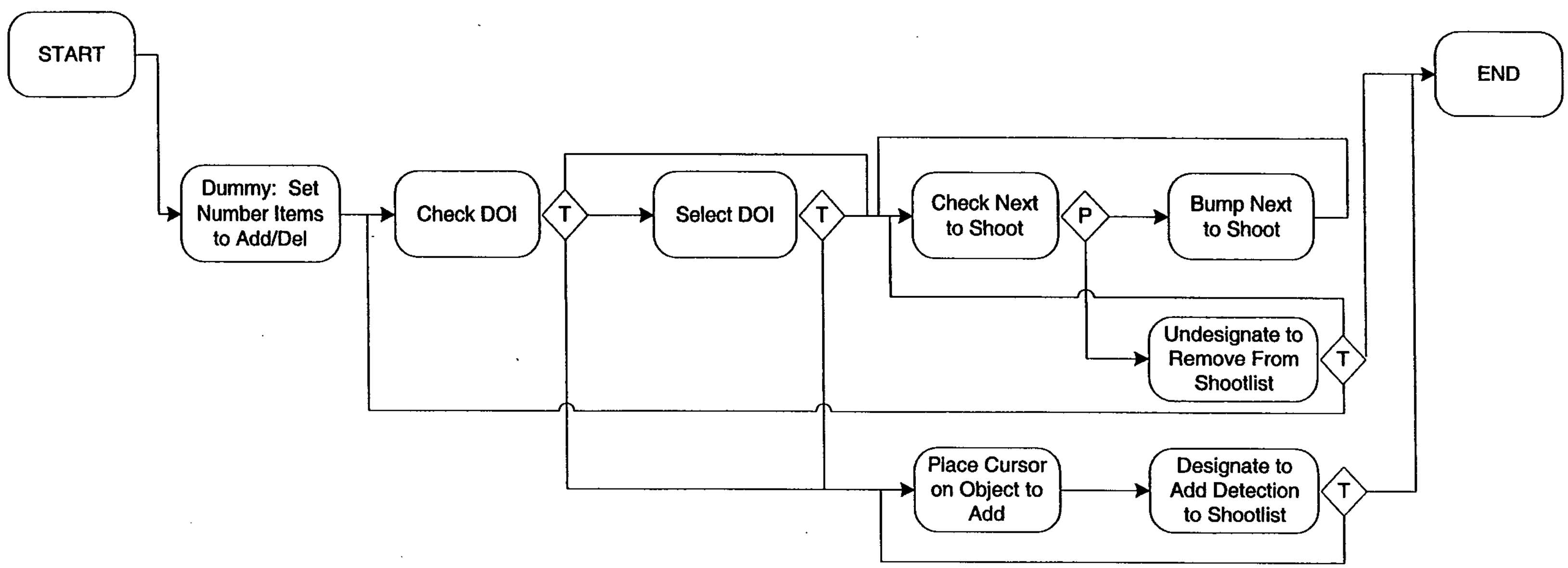


Figure B-25. Task Network for *Update Shootlist* Function Within *Acquire* Goal Function

Description: If after the image is evaluated, the target of interest is not yet identified, the *Update Shootlist* function is used to add or delete objects from the shootlist. The display of interest is set as appropriate, and the number of identified objects (if any) are removed. Similarly, all newly detected (but not identified) objects are added to the shootlist.

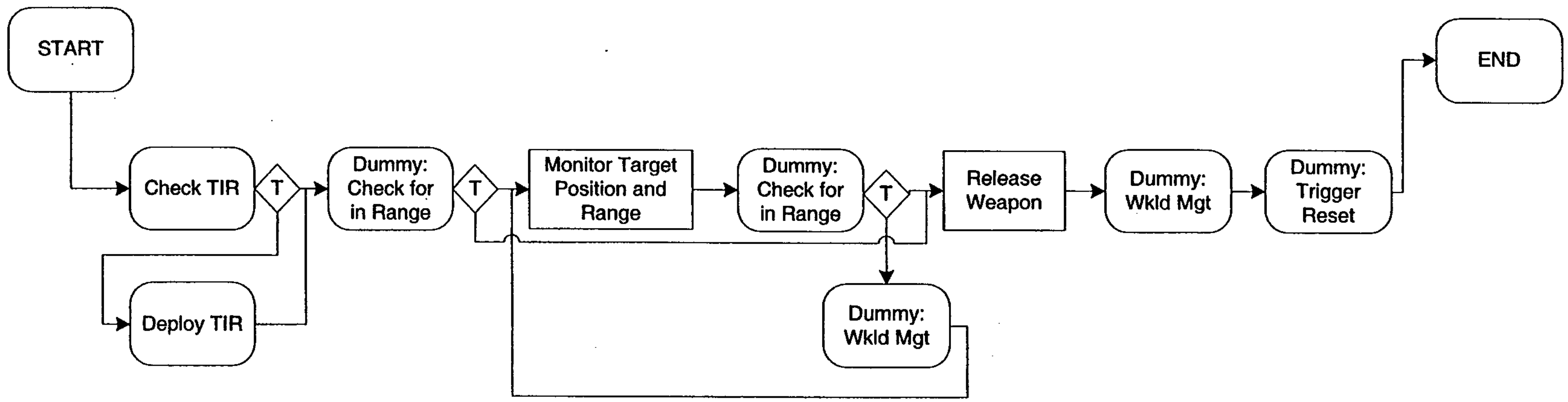


Figure B-26. ***Attack*** **Goal Function Network**

Description: This diagram represents the functions and tasks that support the *Attack* goal function. The TIR is deployed, if necessary, and then a monitoring function checks the range and bearing to the target. Once the target is with weapon release parameters, the weapon is launched. Dummy tasks are used for workload management and to reset triggers.

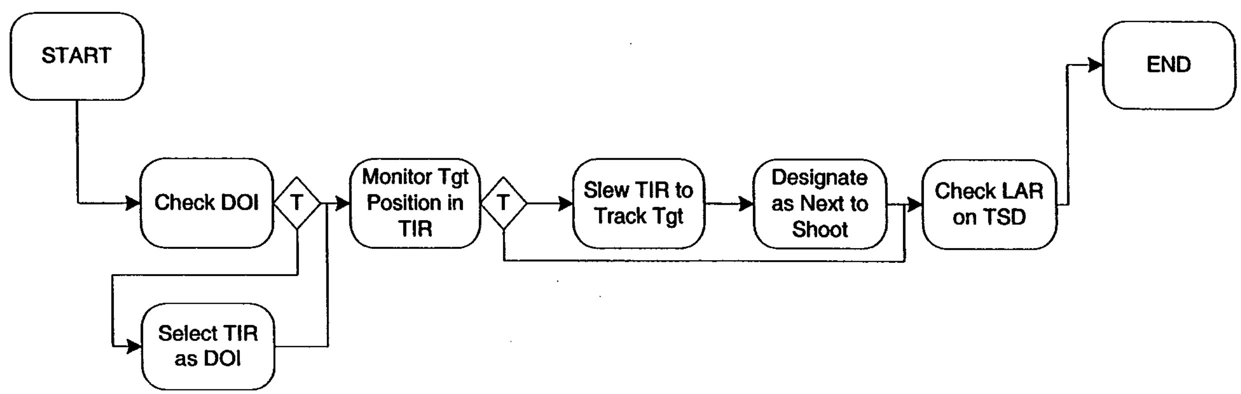


Figure B-27. Task Network for *Monitor Target Position and Range* Function Within *Attack* Goal Function

Description: In this function, the display of interest is set to the TIR, and the range and bearing to the target are monitored. The cursor is slewed to the target and designated as necessary to keep the TIR tracking the target. In addition, the shootlist item (i.e., the target) is checked on the TSD to see if it falls within the LAR (a symbol indicating the weapon release limits).

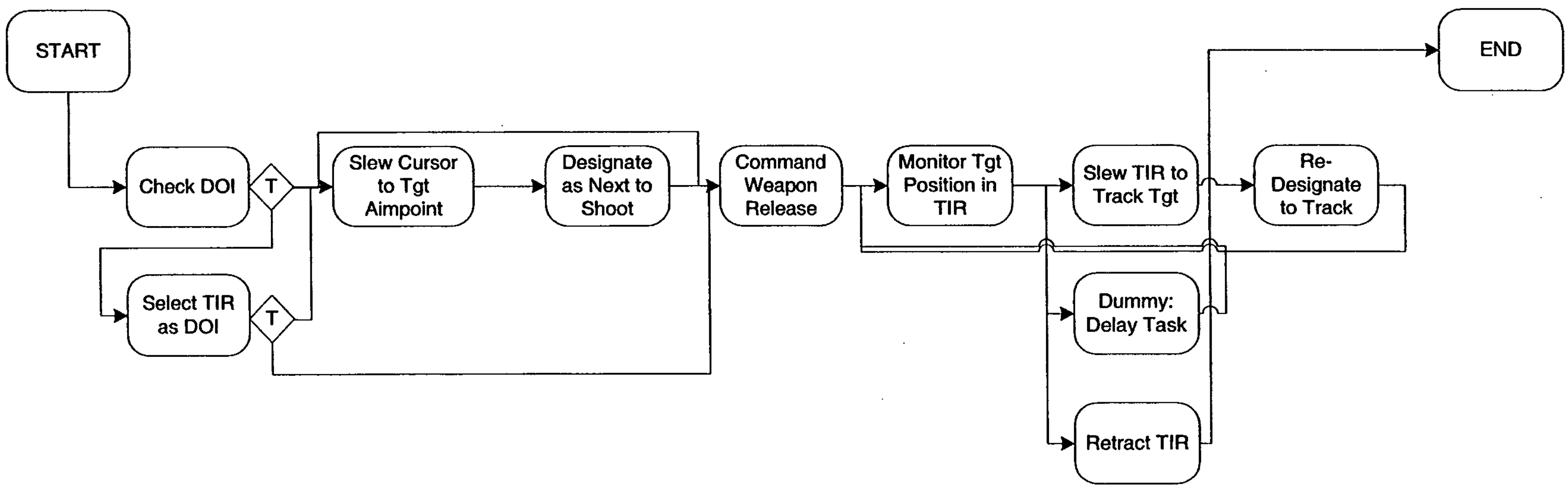


Figure B-28. Task Network for *Release Weapon* Function Within *Attack* Goal Function

Description: The *Weapon Release* function executes once the identified target is in weapon range and bearing. The display of interest is set to the TIR if necessary and the target is redesignated if the TIR has broken track. The weapon is then released. The target continues to be tracked and monitored for a fixed duration of weapon flyout, at which time, the TIR is retracted and the *Attack* goal function ends.

This page left blank intentionally

APPENDIX C

TASK INFORMATION REQUIREMENTS, DECISION DESCRIPTIONS, AND RESULTING COMMANDS IDENTIFIED IN THE MISSION DECOMPOSITION PROCESS

This page left blank intentionally

Table C-1. Results of the Mission Decomposition

Goal	Function Name	Task Name	Information In	Decision	Command Out
Control Aircraft / Maintain SA	Monitor Flight Instruments	Check Airspeed	Current Airspeed		
Control Aircraft / Maintain SA	Monitor Flight Instruments	Check Altitude	Current Altitude		
Control Aircraft / Maintain SA	Monitor Flight Instruments	Check Heading	Current Heading		
Control Aircraft / Maintain SA	Monitor Flight Instruments	Check Attitude	Current Pitch and Roll		
Control Aircraft / Maintain SA	Monitor TSD	Check Mission Leg	Current Destination Point		
Control Aircraft / Maintain SA	Monitor TSD	Check for Threats	Current Number of Launched Missiles		
Control Aircraft / Maintain SA	Monitor TSD	Check Range and Bearing to Target Area	Range and Bearing to Reference Point		
Control Aircraft / Maintain SA	Monitor A/C Systems Status	Check Fuel	Lbs Fuel Remaining		
Control Aircraft / Maintain SA	Monitor Audio	Monitor Audio Channel	Launch Tone Present, Nav Tone Present, Comm Message Present	If tones are present, trigger appropriate goal function (Nav or Evade). If Comm message is present, write down the message.	
Control Aircraft / Maintain SA	Monitor Audio	Receive/Copy Updated Target Coordinates	Updated Target Lat/Lon		
Evade	Monitor Audio	Monitor Audio Channel	Launch Tone Present	If tones are present, trigger appropriate goal functions.	
Evade	Execute Evasion Strategy	Initiate Evade Maneuver			Start Evasive Turn
Evade	Execute Evasion Strategy	Maintain Evade Maneuver	Current heading, desired evasive heading	If current heading = desired evasive heading, then end the turn. Else, continue to turn.	Continue Evasive Turn
Evade	Execute Evasion Strategy	Release Chaff/Flare			Release Chaff/Flare
Evade	Execute Evasion Strategy	Retract TIR			Retract TIR
Evade	Execute Evasion Strategy	Check Countermeasure Stores	Chaff Currently Available	If chaff/flare is available, release it.	
Evade	Execute Evasion Strategy	Move Throttle			Move Throttle to Desired Setting
Evade	Execute Evasion Strategy	Check TIR	TIR Status	If TIR is deployed, then retract it.	
Evade	Execute Evasion Strategy	Check Heading	Current Heading	If current heading = desired heading, then end the turn. Else, continue to turn.	
Evade	Execute Evasion Strategy	Check Attitude	Current Pitch and Roll		
Evade	Execute Evasion Strategy	Check Altitude	Current Altitude		
Evade	Execute Evasion Strategy	Abort Evade Maneuver			Roll Out at Current Heading

Table C-1. Results of the Mission Decomposition (continued)

Goal	Function Name	Task Name	Information In	Decision	Command Out
Evade	Select Evasion Strategy	Determine Maneuver to Intercept Minimization Path	Current Heading, Current Bearing to Threat Emitter	Calculate heading that will steer aircraft 180 deg away from threat emitter location.	
Evade	Evaluate Threat Situation	Identify Current Threat(s)	Current Number of Launched Missiles	If one or more threats are currently launched, prioritize the threats. If threat is no longer active, re-engage autopilot.	
Evade	Evaluate Threat Situation	Prioritize Threats	Missile Type, Range to Emitter, Current Aircraft Altitude	Estimate time to intercept for all active threats and make the nearest threat (in time) the highest priority threat.	
Evade	Re-Engage Autopilot	Engage Autopilot			Engage Autopilot
Evade	Re-Engage Autopilot	Check Autopilot	Current Autopilot Status	If autopilot is not engaged, engage it.	
Evade	Re-Engage Autopilot	Engage Autothrottle			Engage Autothrottle
Evade	Re-Engage Autopilot	Move Throttle			Move Throttle to Desired Setting
Navigate	Plan For Reflying Target Area	Put Cursor on Alt Point			Slew Cursor to Desired Coordinates
Navigate	Plan For Reflying Target Area	Designate as Mustfly Point			Designate Current Cursor Position
Navigate	Plan For Reflying Target Area	Put Cursor on Release Point			Slew Cursor to Desired Coordinates
Navigate	Plan For Reflying Target Area	Select PLAN on UFC Planner Page			Request Re-plan
Navigate	Plan For Reflying Target Area	Evaluate Re-plan	Re-plan is Available, Re-plan Reason	If a re-plan is available and it is not returned as "error" then accept it.	
Navigate	Plan For Reflying Target Area	Accept Re-plan			Accept Re-plan
Navigate	Plan For Attack	Put Cursor Over Target Icon			Slew Cursor to Desired Coordinates
Navigate	Plan For Attack	Designate as Mustfly Point			Designate Current Cursor Position
Navigate	Plan For Attack	Select PLAN on UFC Planner Page			Request Re-plan
Navigate	Plan For Attack	Evaluate Re-plan	Re-plan is Available, Re-plan Reason	If a re-plan is available and it is not returned as "error" then accept it.	
Navigate	Respond to Auto Re-plan	Evaluate Re-plan	Re-plan is Available, Re-plan Reason	If a re-plan is available and it is not returned as "error" then accept it.	
Navigate	Respond to Auto Re-plan	Accept Re-plan			Accept Re-plan
Navigate	Respond to Auto Re-plan	Check DOI	Current Display of Interest, Desired Display of Interest	If current DOI is not the desired DOI, select the desired DOI.	
Navigate	Respond to Auto Re-plan	Select TSD as DOI			Set TSD as Display of Interest
Navigate	Abort	Select ABRT on UFC Planner Page			Request Abort

Table C-1. Results of the Mission Decomposition (continued)

Goal	Function Name	Task Name	Information In	Decision	Command Out
Navigate	Abort	Evaluate Re-plan	Re-plan is Available, Re-plan Reason	If a re-plan is available and it is not returned as "error" then accept it.	
Navigate	Abort	Accept Re-plan			Accept Re-plan
Navigate	Select Planner Mode and DOI	Select Planner Mode on UFC			Select PLAN on UFC
Navigate	Select Planner Mode and DOI	Check UFC Mode	Current UFC Mode	If current UFC mode is not desired UFC mode, select the desired UFC mode.	
Navigate	Select Planner Mode and DOI	Check DOI	Current Display of Interest, Desired Display of Interest	If current DOI is not the desired DOI, select the desired DOI.	
Navigate	Select Planner Mode and DOI	Select TSD as DOI			Select TSD as Display of Interest
Acquire	Evaluate Image	Detect Objects	Entity type, size, and moving status for objects in current sensor field of view	If object can be detected based on Johnson's criteria or GMTI hit, consider it "detected"	
Acquire	Evaluate Image	Identify Objects	Entity type, size, and moving status for objects in current sensor field of view	If object can be identified based on Johnson's criteria, consider it "identified". If object is identified as the target of interest, trigger the appropriate goal function (Nav or Attack).	
Acquire	Update Shootlist	Check DOI	Current Display of Interest, Desired Display of Interest	If current DOI is not the desired DOI, select the desired DOI.	
Acquire	Update Shootlist	Check NTS	Current Shootlist Item Selected		
Acquire	Update Shootlist	Bump NTS			Step Shootlist Item
Acquire	Update Shootlist	Select DOI			Select Desired DOI
Acquire	Update Shootlist	Undesignate to Remove from Shootlist			Remove Item From Shootlist
Acquire	Update Shootlist	Place Cursor on Object to Add			Bump Cursor to Desired Coordinates
Acquire	Update Shootlist	Designate to Add Detection to Shootlist			Designate Current Cursor Position
Acquire	Designate Target	Check DOI	Current Display of Interest	If current DOI is not the desired DOI, select the desired DOI.	
Acquire	Designate Target	Select Radar As DOI			Select Radar as Display of Interest
Acquire	Designate Target	Designate as Next to Shoot			Designate Current Cursor Position
Acquire	Designate Target	Slew Cursor to Target Aimpoint			Slew Cursor to Current Target of Interest Position
Acquire	Designate Target	Select TIR as DOI			Select TIR as Display of Interest

Table C-1. Results of the Mission Decomposition (continued)

Goal	Function Name	Task Name	Information In	Decision	Command Out
Acquire	Update Tgt Location	Check UFC Mode	Current UFC Mode	If current UFC mode is not desired UFC mode, select the desired UFC mode.	
Acquire	Update Tgt Location	Select Target Mode on UFC			Select Target Mode on UFC
Acquire	Update Tgt Location	Enter Updated Tgt Lat/Lon in UFC			Type in Updated Lat/Lon in UFC
Acquire	Update Tgt Location	Accept Updated Tgt Lat/Lon on UFC			Select Accept on UFC
Acquire	Decide Where/How to Look	Estimate Range and Bearing to Ref Point	Range and Bearing to the Reference Point	If reference point is not in range bearing to TIR or SAR, then end acquisition.	
Acquire	Decide Where/How to Look	Choose Sensor and Location to Image	Range and bearing to the reference point for all shootlist items. Moving status of all shootlist items.	If in TIR range and viable objects are on the shootlist, select TIR, next field of view, and highest priority object's coordinates. If in TIR range and no objects on shootlist, select TIR and next alternate look point. If not in TIR range, but in SAR rng/bearing then select radar and next radar lookpoint.	
Acquire	Image Target	Check TIR	Current TIR Status	If next image is TIR and TIR is not deployed, deploy it.	
Acquire	Image Target	Deploy TIR			Deploy TIR
Acquire	Image Target	Retract TIR			Retract TIR
Acquire	Image Target	Check DOI for Designation	Current Display of Interest	If current DOI is not the desired DOI, select the desired DOI.	
Acquire	Image Target	Select DOI for Designation			Select Appropriate DOI (based on sensor chosen)
Acquire	Image Target	Slew Cursor to Alt Look Location			Slew Cursor to Desired Coordinates
Acquire	Image Target	Bump NTS			Step Shootlist Item
Acquire	Image Target	Designate as Next to Shoot			Designate Current Cursor Position
Acquire	Image Target	Select FOV/Command Image			Command Image
Acquire	Image Target	Check UFC Mode	Current UFC Mode	If current UFC mode is not desired UFC mode, select the desired UFC mode.	
Acquire	Image Target	Select Target Mode on UFC			Select Target Mode on UFC
Acquire	Image Target	Check DOI for Imaging	Current Display of Interest	If current DOI is not the desired DOI, select the desired DOI.	
Acquire	Image Target	Select DOI for Imaging			Select Appropriate DOI (based on sensor chosen)

Table C-1. Results of the Mission Decomposition (continued)

Goal	Function Name	Task Name	Information In	Decision	Command Out
Acquire	Image Target	Slew Cursor to Object			Slew Cursor to Desired Location
Acquire		Check TIR	Current TIR Status	If TIR is deployed, retract it.	
Acquire		Retract TIR			Retract TIR
Attack	Monitor Target Position and Range	Check LAR on TSD	Range and Bearing to Designated Point	If target is within LAR, initiate weapon release sequence.	
Attack	Monitor Target Position and Range	Monitor Tgt Position in TIR	Range and Bearing to Designated Point	If target is within LAR, initiate weapon release sequence.	
Attack	Monitor Target Position and Range	Slew TIR to Track Tgt			Slew Cursor to Identified Target of Interest
Attack	Monitor Target Position and Range	Check DOI	Current Display of Interest	If current DOI is not the desired DOI, select the desired DOI.	
Attack	Monitor Target Position and Range	Select TIR as DOI			Select TIR as Display of Interest
Attack	Monitor Target Position and Range	Designate as Next to Shoot			Designate Current Cursor Position
Attack	Release Weapon	Command Weapon Release			Release Weapon
Attack	Release Weapon	Slew Cursor to Target Aimpoint			Slew Cursor to Identified Target of Interest
Attack	Release Weapon	Designate as Next to Shoot			Designate Current Cursor Position
Attack	Release Weapon	Check DOI	Current Display of Interest	If current DOI is not the desired DOI, select the desired DOI.	
Attack	Release Weapon	Select TIR as DOI			Select TIR as Display of Interest
Attack	Release Weapon	Retract TIR			Retract TIR
Attack	Release Weapon	Monitor Tgt Position in TIR	Range and Bearing to Designated Point	If target is within LAR, release weapon.	
Attack	Release Weapon	Slew TIR to Track Tgt			Slew Cursor to Identified Target of Interest
Attack	Release Weapon	Re-Designate to Track			Designate Current Cursor Position
Attack		Check TIR	Current TIR Status	If TIR is retracted, then deploy TIR.	

This page left blank intentionally

APPENDIX D

TABLES OF PERFORMANCE TIMES AND WORKLOAD VALUES ASSIGNED TO MODEL TASKS

This page left blank intentionally

Table D-1. Performance Times and Workload Values[1] Assigned to Model Tasks

Goal	Function	Task	Mean Time	V	A	C	P
Acquire	Decide Where/How to Look	Choose Sensor and Location to Image	dec_time/60;	5.0	0.0	6.8	0.0
Acquire	Decide Where/How to Look	Estimate Range and Bearing to Ref Point	0:00:01.00	5.0	0.0	4.6	0.0
Acquire	Designate Target	Check DOI	0:00:00.50	4.0	0.0	1.0	0.0
Acquire	Designate Target	Designate as Next to Shoot	0:00:00.40	0.0	0.0	1.0	2.2
Acquire	Designate Target	Select Radar As DOI	0:00:00.90	1.0	0.0	1.0	2.2
Acquire	Designate Target	Select TIR as DOI	0:00:00.90	1.0	0.0	1.0	2.2
Acquire	Designate Target	Slew Cursor to Target Aimpoint	0:00:02.00	5.4	0.0	1.0	5.8
Acquire	Evaluate Image	Detect Objects	0:00:02.00	7.0	0.0	3.7	0.0
Acquire	Evaluate Image	Identify Objects	0:00:02.00	7.0	0.0	6.8	0.0
Acquire	Image Target	Bump NTS	0.6*(pl_number/2)/60;	4.0	0.0	1.2	2.2
Acquire	Image Target	Check DOI for Designation	0:00:00.50	4.0	0.0	1.0	0.0
Acquire	Image Target	Check DOI for Imaging	0:00:00.50	4.0	0.0	1.0	0.0
Acquire	Image Target	Check TIR	0:00:00.50	4.0	0.0	1.0	0.0
Acquire	Image Target	Check UFC Mode	0:00:00.70	4.0	0.0	1.0	0.0
Acquire	Image Target	Deploy TIR	0:00:00.90	4.0	0.0	1.0	2.2
Acquire	Image Target	Designate as Next to Shoot	0:00:00.40	0.0	0.0	1.0	2.2
Acquire	Image Target	Retract TIR	0:00:00.90	4.0	0.0	1.0	2.2
Acquire	Image Target	Select DOI for Designation	0:00:00.90	1.0	0.0	1.0	2.2
Acquire	Image Target	Select DOI for Imaging	0:00:00.90	1.0	0.0	1.0	2.2
Acquire	Image Target	Select FOV/Command Image	task_time/60;	1.0	0.0	1.0	2.2
Acquire	Image Target	Select Target Mode on UFC	0:00:01.08	1.0	0.0	1.0	2.2
Acquire	Image Target	Slew Cursor to Alt Look Location	0:00:02.00	5.4	0.0	1.0	5.8
Acquire	Image Target	Slew Cursor to Object	0:00:02.00	5.4	0.0	1.0	5.8
Acquire	Update Shootlist	Bump NTS	0:00:00.60	4.0	0.0	1.2	2.2
Acquire	Update Shootlist	Check DOI	0:00:00.50	4.0	0.0	1.0	0.0
Acquire	Update Shootlist	Check NTS	0:00:00.50	4.0	0.0	1.2	0.0
Acquire	Update Shootlist	Designate to Add Detection to Shootlist	0:00:00.40	0.0	0.0	1.0	2.2
Acquire	Update Shootlist	Place Cursor on Object to Add	0:00:02.00	5.0	0.0	1.0	5.8
Acquire	Update Shootlist	Select DOI	0:00:00.90	1.0	0.0	1.0	2.2
Acquire	Update Shootlist	Undesignate to Remove from Shootlist	0:00:00.40	0.0	0.0	1.0	2.2
Acquire	Update Tgt Location	Accept Updated Tgt Lat/Lon on UFC	0:00:00.40	1.0	0.0	1.0	2.2
Acquire	Update Tgt Location	Check UFC Mode	0:00:00.70	4.0	0.0	1.0	0.0
Acquire	Update Tgt Location	Enter Updated Tgt Lat/Lon in UFC	0:00:15.00	5.9	0.0	5.3	7.0
Acquire	Update Tgt Location	Select Target Mode on UFC	0:00:01.08	1.0	0.0	1.0	2.2
Acquire		Check TIR	0:00:00.50	4.0	0.0	1.0	0.0

[1] VACP values in table represent visual, auditory, cognitive and psychomotor dimensions, respectively.

Table D-1. Performance Times and Workload Values Assigned to Model Tasks (continued)

Goal	Function	Task	Mean Time	V	A	C	P
Acquire		Check TIR	0:00:00.50	4.0	0.0	1.0	0.0
Acquire		Deploy TIR	0:00:00.90	4.0	0.0	1.0	2.2
Acquire		Retract TIR	0:00:00.90	0.0	0.0	1.0	2.2
Attack	Monitor Target Position and Range	Check DOI	0:00:00.50	4.0	0.0	1.0	0.0
Attack	Monitor Target Position and Range	Check LAR on TSD	0:00:00.70	5.0	0.0	4.6	0.0
Attack	Monitor Target Position and Range	Designate as Next to Shoot	0:00:00.40	0.0	0.0	1.0	2.2
Attack	Monitor Target Position and Range	Monitor Tgt Position in TIR	0:00:00.70	4.0	0.0	1.0	0.0
Attack	Monitor Target Position and Range	Select TIR as DOI	0:00:00.90	1.0	0.0	1.0	2.2
Attack	Monitor Target Position and Range	Slew TIR to Track Tgt	0:00:02.00	5.4	0.0	1.0	5.8
Attack	Release Weapon	Check DOI	0:00:00.50	4.0	0.0	1.0	0.0
Attack	Release Weapon	Command Weapon Release	0:00:00.40	0.0	0.0	1.0	2.2
Attack	Release Weapon	Designate as Next to Shoot	0:00:00.40	0.0	0.0	1.0	2.2
Attack	Release Weapon	Monitor Tgt Position in TIR	0:00:00.70	4.0	0.0	1.0	0.0
Attack	Release Weapon	Re-Designate to Track	0:00:00.40	0.0	0.0	1.0	2.2
Attack	Release Weapon	Retract TIR	0:00:00.90	0.0	0.0	1.0	2.2
Attack	Release Weapon	Select TIR as DOI	0:00:00.90	1.0	0.0	1.0	2.2
Attack	Release Weapon	Slew Cursor to Target Aimpoint	0:00:02.00	5.4	0.0	1.0	5.8
Attack	Release Weapon	Slew TIR to Track Tgt	0:00:02.00	5.4	0.0	1.0	5.8
Evade	Evaluate Threat Situation	Identify Current Threat(s)	0:00:00.70	5.9	0.0	5.3	0.0
Evade	Evaluate Threat Situation	Prioritize Threats	(0.31/60)*(tht_count * (tht_count-1)/2);	5.0	0.0	7.0	0.0
Evade	Execute Evasion Strategy	Check Altitude	0:00:00.50	5.9	0.0	3.7	0.0
Evade	Execute Evasion Strategy	Check Attitude	0:00:00.50	5.0	0.0	1.0	0.0
Evade	Execute Evasion Strategy	Check Countermeasure Stores	0:00:00.70	5.9	0.0	3.7	0.0
Evade	Execute Evasion Strategy	Check Heading	0:00:00.50	5.0	0.0	4.6	0.0
Evade	Execute Evasion Strategy	Check TIR	0:00:00.50	4.0	0.0	1.0	0.0
Evade	Execute Evasion Strategy	Initiate Evade Maneuver	0:00:00.50	5.4	0.0	4.6	2.6
Evade	Execute Evasion Strategy	Maintain Evade Maneuver	0:00:00.50	0.0	0.0	0.0	2.6
Evade	Execute Evasion Strategy	Move Throttle	0:00:00.50	0.0	0.0	1.0	5.8
Evade	Execute Evasion Strategy	Release Chaff/Flare	0:00:00.40	0.0	0.0	1.0	2.2
Evade	Execute Evasion Strategy	Retract TIR	0:00:00.90	0.0	0.0	1.0	2.2
Evade	Monitor Audio	Monitor Audio Channel	0:00:00.50	0.0	1.0	1.0	0.0
Evade	Re-Engage Autopilot	Check Autopilot	0:00:00.70	4.0	0.0	1.0	0.0
Evade	Re-Engage Autopilot	Engage Autopilot	0:00:01.40	0.0	0.0	1.0	2.2
Evade	Re-Engage Autopilot	Engage Autothrottle	0:00:01.40	0.0	0.0	1.0	2.2

Table D-1. Performance Times and Workload Values Assigned to Model Tasks (continued)

Goal	Function	Task	Mean Time	V	A	C	P
Evade	Re-Engage Autopilot	Move Throttle	0:00:01.00	5.0	0.0	1.0	5.8
Evade	Select Evasion Strategy	Determine Jink	0:00:00.07	5.0	0.0	1.2	0.0
Evade	Select Evasion Strategy	Determine Maneuver to Intercept Minimization Path	0:00:00.07	5.0	0.0	6.8	0.0
Mission Level	Monitor A/C Systems Status	Check Fuel	0:00:00.70	5.9	0.0	3.7	0.0
Mission Level	Monitor Audio	Monitor Audio Channel	0:00:00.50	0.0	1.0	1.0	0.0
Mission Level	Monitor Audio	Receive/Copy Updated Target Coordinates	0:00:23.00	5.9	4.9	5.3	6.5
Mission Level	Monitor Flight Instruments	Check Airspeed	0:00:00.70	5.9	0.0	3.7	0.0
Mission Level	Monitor Flight Instruments	Check Altitude	0:00:00.50	5.9	0.0	3.7	0.0
Mission Level	Monitor Flight Instruments	Check Attitude	0:00:00.50	5.0	0.0	1.0	0.0
Mission Level	Monitor Flight Instruments	Check Heading	0:00:00.50	5.0	0.0	4.6	0.0
Mission Level	Monitor TSD	Check for Threats	0:00:00.50	4.0	0.0	1.0	0.0
Mission Level	Monitor TSD	Check Mission Leg	0:00:00.70	4.0	0.0	1.0	0.0
Mission Level	Monitor TSD	Check Range and Bearing to Target Area	0:00:01.00	5.0	0.0	4.6	0.0
Navigate	Abort	Accept Re-plan	0:00:01.80	0.0	0.0	1.0	2.2
Navigate	Abort	Evaluate Re-plan	0:00:01.27	7.0	0.0	6.8	0.0
Navigate	Abort	Select ABRT on UFC Planner Page	0:00:01.08	1.0	0.0	1.0	2.2
Navigate	Plan For Attack	Accept Re-plan	0:00:01.80	1.0	0.0	1.0	2.2
Navigate	Plan For Attack	Designate as Mustfly Point	0:00:00.40	0.0	0.0	1.0	2.2
Navigate	Plan For Attack	Evaluate Re-plan	0:00:01.77	7.0	0.0	6.8	0.0
Navigate	Plan For Attack	Put Cursor Over Target Icon	0:00:02.00	5.0	0.0	1.0	5.8
Navigate	Plan For Attack	Select PLAN on UFC Planner Page	0:00:01.08	1.0	0.0	1.0	2.2
Navigate	Plan For Reflying Target Area	Accept Re-plan	0:00:01.80	1.0	0.0	1.0	2.2
Navigate	Plan For Reflying Target Area	Designate as Mustfly Point	0:00:00.40	0.0	0.0	1.0	2.2
Navigate	Plan For Reflying Target Area	Evaluate Re-plan	0:00:01.77	7.0	0.0	6.8	0.0
Navigate	Plan For Reflying Target Area	Put Cursor on Alt Point	0:00:02.00	5.0	0.0	1.0	5.8
Navigate	Plan For Reflying Target Area	Put Cursor on Release Point	0:00:02.00	5.0	0.0	1.0	5.8
Navigate	Plan For Reflying Target Area	Select PLAN on UFC Planner Page	0:00:01.08	1.0	0.0	1.0	2.2
Navigate	Respond to Auto Re-plan	Accept Re-plan	0:00:00.40	1.0	0.0	1.0	2.2
Navigate	Respond to Auto Re-plan	Check DOI	0:00:00.50	4.0	0.0	1.0	0.0
Navigate	Respond to Auto Re-plan	Evaluate Re-plan	0:00:01.77	7.0	0.0	6.8	0.0
Navigate	Respond to Auto Re-plan	Select TSD as DOI	0:00:00.90	1.0	0.0	1.0	2.2
Navigate	Select Planner Mode and DOI	Check DOI	0:00:00.50	4.0	0.0	1.0	0.0

Table D-1. Performance Times and Workload Values Assigned to Model Tasks (continued)

Goal	Function	Task	Mean Time	V	A	C	P
Navigate	Select Planner Mode and DOI	Check UFC Mode	0:00:00.70	4.0	0.0	1.0	0.0
Navigate	Select Planner Mode and DOI	Select Planner Mode on UFC	0:00:01.08	1.0	0.0	1.0	2.2
Navigate	Select Planner Mode and DOI	Select TSD as DOI	0:00:00.90	1.0	0.0	1.0	2.2

APPENDIX E

VARIABLES USED IN CASE STUDY 1 HUMAN PERFORMANCE MODEL

This page left blank intentionally

Table E-1. Variables Used in Case Study 1 Human Performance Model

Variable Name	Variable Type	External
abort_done	Integer	FALSE
abs_turn	Real	FALSE
add_sl_count	Integer	FALSE
AFTERBURNER	Integer	FALSE
alt_pt_index	Integer	FALSE
alt_pt_inuse	Integer	FALSE
alt_pt_used	Array of Integers	FALSE
alt_pts_done	Integer	FALSE
atk_rpln_don	Integer	FALSE
attack_done	Integer	FALSE
audio_done	Integer	FALSE
auto_pilot	Integer	FALSE
AUTOTHROTTLE	Integer	FALSE
bad_tht	Array of Integers	FALSE
check_upd	Integer	FALSE
cmd_snsr_cl	Integer	FALSE
cns_moving	Integer	FALSE
count	Integer	FALSE
CRUISE	Integer	FALSE
ct	Integer	FALSE
cur_throttle	Integer	FALSE
curr_emitter	Integer	FALSE
current_doi	Integer	FALSE
dec_time	Real	FALSE
del_sl_count	Integer	FALSE
detect_il	Array of Integers	FALSE
done	Integer	FALSE
end_acq	Integer	FALSE
end_audio	Integer	FALSE
end_evd_wkld	Integer	FALSE
end_turn	Integer	FALSE
eval_gmt_hit	Integer	FALSE
eval_moving	Integer	FALSE
eval_rng	Real	FALSE
eval_size	Real	FALSE
eval_type	Integer	FALSE
found	Integer	FALSE
found_ptr	Integer	FALSE
found_status	Integer	FALSE
FULL_MIL	Integer	FALSE
give_up	Integer	FALSE
halt_done	Integer	FALSE
id_to_image	Integer	FALSE
ident_as_tgt	Integer	FALSE
idx	Integer	FALSE
in_sar_rng	Integer	FALSE
in_snsr_rng	Integer	FALSE

Table E-1. Variables Used in Case Study 1 Human Performance Model (continued)

Variable Name	Variable Type	External
in_tir_rng	Integer	FALSE
index	Integer	FALSE
index_to_alt	Integer	FALSE
index_to_pl	Integer	FALSE
index_wrap	Integer	FALSE
input_brg	Real	FALSE
input_il_idx	Integer	FALSE
input_pl_idx	Integer	FALSE
input_rng	Real	FALSE
input_sensor	Integer	FALSE
int_min_hdg	Real	FALSE
K_A_PILOT_D	Integer	FALSE
K_A_PILOT_E	Integer	FALSE
K_A_THROTTLE	Integer	FALSE
K_ALT_LK_LAT	Array of Reals	FALSE
K_ALT_LK_LON	Array of Reals	FALSE
K_ALT_PT_LAT	Array of Reals	FALSE
K_ALT_PT_LON	Array of Reals	FALSE
K_BINGO_FUEL	Real	FALSE
K_BLD_AIRFLD	Integer	FALSE
K_BLD_BRIDGE	Integer	FALSE
K_BLD_COM	Integer	FALSE
K_BLD_MFG	Integer	FALSE
K_BLD_PWR	Integer	FALSE
K_BLD_SAM	Integer	FALSE
K_CALC_ALTPT	Integer	FALSE
K_CURSOR_AC	Integer	FALSE
K_DELAY_END	Integer	FALSE
K_DEPLOY_TIR	Integer	FALSE
K_DESIG_PT	Integer	FALSE
K_DETECTED	Integer	FALSE
K_DFLT_SNSR	Integer	FALSE
K_DOI_CENTER	Integer	FALSE
K_DOI_CTRMPD	Integer	FALSE
K_DOI_LEFT	Integer	FALSE
K_DOI_LFTMPD	Integer	FALSE
K_DOI_RIGHT	Integer	FALSE
K_DOI_RTMPD	Integer	FALSE
K_EMT_TYPE_1	Integer	FALSE
K_EMT_TYPE_2	Integer	FALSE
K_EVADE_MNVR	Integer	FALSE
K_F_ACP_RPLN	Integer	FALSE
K_GAMEOVER	Integer	FALSE
K_GEN_IMAGE	Integer	FALSE
K_GND_BRDM	Integer	FALSE

Table E-1. Variables Used in Case Study 1 Human Performance Model (continued)

Variable Name	Variable Type	External
K_GND_BTR	Integer	FALSE
K_GND_M1A1	Integer	FALSE
K_GND_MTLB	Integer	FALSE
K_GND_SCUD	Integer	FALSE
K_GND_T72	Integer	FALSE
K_GND_T90	Integer	FALSE
K_GND_TAPZ	Integer	FALSE
K_GND_TRACKD	Integer	FALSE
K_GND_TRUCK	Integer	FALSE
K_GND_VAN	Integer	FALSE
K_HDG_ADJ	Real	FALSE
K_HOME_LAT	Real	FALSE
K_HOME_LON	Real	FALSE
K_IDENTIFIED	Integer	FALSE
K_JINK_LEFT	Integer	FALSE
K_JINK_RIGHT	Integer	FALSE
K_LAR_RNG	Real	FALSE
K_LATLON_ADJ	Real	FALSE
K_LK_PT_LAT	Array of Reals	FALSE
K_LK_PT_LON	Array of Reals	FALSE
K_MAK_MSTFLY	Integer	FALSE
K_MAX_ALT_LK	Integer	FALSE
K_MOVECURSOR	Integer	FALSE
K_NSAR_HIGH	Integer	FALSE
K_NSAR_LOW	Integer	FALSE
K_NSAR_MED	Integer	FALSE
K_NUM_ALT_PT	Array of Integers	FALSE
K_ORIGWR_LAT	Array of Reals	FALSE
K_ORIGWR_LON	Array of Reals	FALSE
K_PROCES_OOI	Integer	FALSE
K_PX_FACTOR	Integer	FALSE
K_RANK_OOI	Integer	FALSE
K_RBEAM_ROI	Integer	FALSE
K_REAL_BEAM	Integer	FALSE
K_RECALC_PL	Integer	FALSE
K_REL_CMEAS	Integer	FALSE
K_REL_WCM	Integer	FALSE
K_REMOVE_PL	Integer	FALSE
K_REQ_ABORT	Integer	FALSE
K_REQ_RPLN	Integer	FALSE
K_RESET_OOI	Integer	FALSE
K_RESET_SL	Integer	FALSE
K_RMV_SL_OBJ	Integer	FALSE
K_RP_TYP_ERR	Integer	FALSE
K_RP_TYP_FUL	Integer	FALSE
K_RP_TYP_NAV	Integer	FALSE

Table E-1. Variables Used in Case Study 1 Human Performance Model (continued)

Variable Name	Variable Type	External
K_RP_TYP_REQ	Integer	FALSE
K_RP_TYP_THT	Integer	FALSE
K_S_LP_IDX	Array of Integers	FALSE
K_S_LP_MAX	Integer	FALSE
K_S_LP_SNSR	Array of Integers	FALSE
K_SAR_GIM_HI	Real	FALSE
K_SAR_GIM_LO	Real	FALSE
K_SAR_RNG	Real	FALSE
K_STOW_TIR	Integer	FALSE
K_T_LP_IDX	Array of Integers	FALSE
K_T_LP_MAX	Integer	FALSE
K_T_LP_SNSR	Array of Integers	FALSE
K_TGTBEG_LAT	Array of Reals	FALSE
K_TGTBEG_LON	Array of Reals	FALSE
K_THROTTLE	Integer	FALSE
K_TIR_2X_NAR	Integer	FALSE
K_TIR_NARROW	Integer	FALSE
K_TIR_RNG	Integer	FALSE
K_TIR_WIDE	Integer	FALSE
K_U_ACP_RPLN	Integer	FALSE
K_UFC_LL_ACP	Integer	FALSE
K_UFC_NAV	Integer	FALSE
K_UFC_PLAN	Integer	FALSE
K_UFC_PLNMOD	Integer	FALSE
K_UFC_TGT	Integer	FALSE
K_UFC_TGT_LL	Integer	FALSE
K_UFC_TGTMOD	Integer	FALSE
K_UFC_THT	Integer	FALSE
K_UNVIABLE	Integer	FALSE
K_UPD_REFLOC	Integer	FALSE
K_UPD_SENSOR	Integer	FALSE
K_WSAR_HIGH	Integer	FALSE
K_WSAR_LOW	Integer	FALSE
K_WSAR_MED	Integer	FALSE
L_LK_PT_LAT	Integer	FALSE
l_num_iar	Integer	FALSE
last_apd_cnt	Integer	FALSE
last_rti_trg	Integer	FALSE
last_sensor	Integer	FALSE
last_snsr_cl	Integer	FALSE
last_tir_id	Integer	FALSE
lastpl_count	Integer	FALSE
lat_to_image	Real	FALSE
lon_to_image	Real	FALSE
must_evade	Integer	FALSE

Table E-1. Variables Used in Case Study 1 Human Performance Model (continued)

Variable Name	Variable Type	External
nav_abrt_pdg	Integer	FALSE
nav_fnd_pdg	Integer	FALSE
nav_nfnd_pdg	Integer	FALSE
need_navplan	Integer	FALSE
need_planchk	Integer	FALSE
need_reflook	Integer	FALSE
next_doi	Integer	FALSE
next_isar_lp	Integer	FALSE
next_itir_lp	Integer	FALSE
next_jink	Integer	FALSE
next_usar_lp	Integer	FALSE
next_utir_lp	Integer	FALSE
nxtsnsr	Integer	FALSE
obj_angle	Real	FALSE
obj_detected	Integer	FALSE
obj_ident	Integer	FALSE
obj_is_toi	Integer	FALSE
OFF	Integer	FALSE
ON	Integer	FALSE
orig_wrp_act	Integer	FALSE
orig_wrp_id	Integer	FALSE
p_ac_lat	Real	FALSE
p_ac_lon	Real	FALSE
p_airspeed	Real	FALSE
p_altitude	Real	FALSE
p_brg_to_dp	Real	FALSE
p_brg_to_ref	Real	FALSE
p_chf_avail	Integer	FALSE
p_curr_doi	Integer	FALSE
p_dest_lat	Real	FALSE
p_dest_lon	Real	FALSE
p_flr_avail	Integer	FALSE
p_fuel_qty	Real	FALSE
p_heading	Integer	FALSE
p_launch_ton	Integer	FALSE
p_pitch	Real	FALSE
p_power	Real	FALSE
p_rng_to_dp	Real	FALSE
p_rng_to_ref	Real	FALSE
p_roll	Real	FALSE
p_rpln_avail	Integer	FALSE
p_rpln_reasn	Integer	FALSE
p_rpln_ton	Integer	FALSE
p_tgt_update	Integer	FALSE

Table E-1. Variables Used in Case Study 1 Human Performance Model (continued)

Variable Name	Variable Type	External
p_tht_count	Integer	FALSE
p_tht_emtbrg	Array of Reals	FALSE
p_tht_emtrng	Array of Reals	FALSE
p_tht_emttyp	Array of Integers	FALSE
p_tht_msltti	Array of Reals	FALSE
p_ufc_mode	Integer	FALSE
pass	Integer	FALSE
pass_clock	Real	FALSE
pl_in_rng	Integer	FALSE
plist_count	Integer	FALSE
pre_tir_pndg	Integer	FALSE
pre_tir_upd	Integer	FALSE
prev_emitter	Integer	FALSE
prev_img_id	Integer	FALSE
pri_number	Integer	FALSE
ptr	Integer	FALSE
px_displayed	Real	FALSE
px_to_detect	Real	FALSE
px_to_ident	Real	FALSE
recording	Integer	FALSE
ref_adjust	Integer	FALSE
ref_in_rng	Integer	FALSE
replan_timer	Real	FALSE
rp_next_step	Integer	FALSE
SAR	Integer	FALSE
sensor	Integer	FALSE
shortest_tti	Real	FALSE
shotdown	Integer	FALSE
slew_needed	Integer	FALSE
start_timer	Real	FALSE
tan_angle	Real	FALSE
task_time	Real	FALSE
temp_hdg	Real	FALSE
tgt_recorded	Integer	FALSE
tgt_upd_done	Integer	FALSE
tht_index	Integer	FALSE
TIR	Integer	FALSE
tir_deployed	Integer	FALSE
tir_fov	Integer	FALSE
toi_found	Integer	FALSE
toi_index	Integer	FALSE
top_threat	Integer	FALSE
trg_acq_rb	Integer	FALSE
trg_acq_upd	Integer	FALSE

Table E-1. Variables Used in Case Study 1 Human Performance Model (continued)

Variable Name	Variable Type	External
trg_attack	Integer	FALSE
trg_evade	Integer	FALSE
trg_nav_abrt	Integer	FALSE
trg_nav_fnd	Integer	FALSE
trg_nav_nfnd	Integer	FALSE
trg_nav_pln	Integer	FALSE
trg_nav_rte	Integer	FALSE
turn	Real	FALSE
ufc_mode	Integer	FALSE
upd_ref_lat	Real	FALSE
upd_ref_lon	Real	FALSE
upd_wrp_lat	Real	FALSE
upd_wrp_lon	Real	FALSE
WKLD_ACQ	Integer	FALSE
WKLD_ATK	Integer	FALSE
WKLD_FTP	Integer	FALSE
WKLD_NAV	Integer	FALSE
wpn_clock	Real	FALSE
wpt_tmpclock	Real	FALSE
zoom_only	Integer	FALSE
FALSE	Integer	FALSE
TRUE	Integer	FALSE
ac_airspeed	Real	TRUE
ac_altitude	Real	TRUE
ac_fuel_qty	Real	TRUE
ac_heading	Real	TRUE
ac_lat	Real	TRUE
ac_lon	Real	TRUE
ac_pitch	Real	TRUE
ac_power	Integer	TRUE
ac_roll	Real	TRUE
ac_shotdown	Integer	TRUE
apd_brg	Real	TRUE
apd_cnt	Integer	TRUE
apd_rng	Real	TRUE
brg_to_dp	Real	TRUE
brg_to_ref	Real	TRUE
chf_avail	Integer	TRUE
desig_elev	Real	TRUE
desig_lat	Real	TRUE
desig_lon	Real	TRUE
dest_pt_id	Integer	TRUE
dest_pt_lat	Real	TRUE
dest_pt_lon	Real	TRUE
flr_avail	Integer	TRUE

Table E-1. Variables Used in Case Study 1 Human Performance Model (continued)

Variable Name	Variable Type	External
hpm_alt_lat	Real	TRUE
hpm_alt_lon	Real	TRUE
hpm_cmd_id	Integer	TRUE
hpm_curs_lat	Real	TRUE
hpm_curs_lon	Real	TRUE
hpm_des_hdg	Real	TRUE
hpm_des_thtl	Integer	TRUE
hpm_des_turn	Real	TRUE
hpm_ref_lat	Real	TRUE
hpm_ref_lon	Real	TRUE
hpm_rti_trig	Integer	TRUE
hpm_upd_lat	Real	TRUE
hpm_upd_lon	Real	TRUE
iar_elev	Array of Reals	TRUE
iar_num_det	Integer	TRUE
iar_num_id	Integer	TRUE
iar_num_obj	Integer	TRUE
iar_obj_id	Array of Integers	TRUE
iar_obj_lat	Array of Reals	TRUE
iar_obj_lon	Array of Reals	TRUE
iar_obj_mvng	Array of Integers	TRUE
iar_obj_nxts	Array of Integers	TRUE
iar_obj_stat	Array of Integers	TRUE
iar_rti_trig	Integer	TRUE
iar_sensor	Integer	TRUE
il_elev	Array of Reals	TRUE
il_gmti_det	Array of Integers	TRUE
il_image_id	Integer	TRUE
il_lat	Array of Reals	TRUE
il_lon	Array of Reals	TRUE
il_mover_idx	Array of Integers	TRUE
il_moving	Array of Integers	TRUE
il_number	Integer	TRUE
il_obj_id	Array of Integers	TRUE
il_prev_det	Array of Integers	TRUE
il_prev_id	Array of Integers	TRUE
il_range	Array of Reals	TRUE
il_rng_ref	Array of Reals	TRUE
il_sensor	Integer	TRUE
il_size	Array of Reals	TRUE
il_type	Array of Integers	TRUE
launch_ton	Integer	TRUE
mp_id	Array of Integers	TRUE
mp_idx_2_tbm	Integer	TRUE

Table E-1. Variables Used in Case Study 1 Human Performance Model (continued)

Variable Name	Variable Type	External
mp_lat	Array of Reals	TRUE
mp_lon	Array of Reals	TRUE
mp_num	Integer	TRUE
obj_rng	Array of Reals	TRUE
pl_brg_ac	Array of Reals	TRUE
pl_count	Integer	TRUE
pl_id	Array of Integers	TRUE
pl_lat	Array of Reals	TRUE
pl_lon	Array of Reals	TRUE
pl_num_add	Integer	TRUE
pl_num_del	Integer	TRUE
pl_number	Integer	TRUE
pl_nxtsensor	Array of Integers	TRUE
pl_rng_ac	Array of Reals	TRUE
pl_rng_ref	Array of Reals	TRUE
pl_type	Array of Integers	TRUE
pl_viable	Integer	TRUE
rng_to_dp	Real	TRUE
rng_to_ref	Real	TRUE
route	Integer	TRUE
rpln_avail	Integer	TRUE
rpln_count	Integer	TRUE
rpln_reason	Integer	TRUE
rpln_ton	Integer	TRUE
sensor_2_use	Integer	TRUE
tgt_update	Integer	TRUE
tht_count	Integer	TRUE
tht_emtbrg	Array of Reals	TRUE
tht_emtid	Array of Integers	TRUE
tht_emtrng	Array of Reals	TRUE
tht_emttyp	Array of Integers	TRUE
tht_mslid	Array of Integers	TRUE
tht_msltti	Array of Reals	TRUE
tir_dep_stat	Integer	TRUE
tir_trk_stat	Integer	TRUE
updated_lat	Real	TRUE
updated_lon	Real	TRUE

This page left blank intentionally

APPENDIX F

MODEL CODE RELEASE CONDITIONS, BEGINNING EFFECTS, ENDING EFFECTS, AND DECISION NODES

This page left blank intentionally

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Control Aircraft / Maintain SA		START						
Control Aircraft / Maintain SA	Monitor Flight Instruments	START						
Control Aircraft / Maintain SA	Monitor Flight Instruments	Check Airspeed	(WKLD_NAV==FALSE & WKLD_ACQ==FALSE & WKLD_ATK==FALSE & WKLD_FTP==FALSE & curgolstatus[1]<>1) ;	WKLD_FTP:=TRUE;	p_airspeed:=ac_airspeed;p_power:=ac_power;			
Control Aircraft / Maintain SA	Monitor Flight Instruments	Check Altitude			p_altitude:=ac_altitude;			
Control Aircraft / Maintain SA	Monitor Flight Instruments	Check Heading			p_heading:=ac_heading;			
Control Aircraft / Maintain SA	Monitor Flight Instruments	Check Attitude			p_roll:=ac_roll;p_pitch:=ac_pitch;			
Control Aircraft / Maintain SA	Monitor Flight Instruments	END						
Control Aircraft / Maintain SA	Monitor TSD	START						
Control Aircraft / Maintain SA	Monitor TSD	Check Mission Leg		p_ac_lat:=ac_lat;p_ac_lon:=ac_lon;{if we haven't found the target or decided to abort (both of which set rp_next_step = 99) then call macro to decide whether to refly (1st pass) or abort (2nd pass) if target is not found}if rp_next_step<>99 then CHOOSE_DEST;				
Control Aircraft / Maintain SA	Monitor TSD	Check for Threats			p_tht_count:=tht_count;			
Control Aircraft / Maintain SA	Monitor TSD	Check Range and Bearing to Target Area		p_rng_to_ref:=rng_to_ref;p_brg_to_ref:=brg_to_ref;p_rng_to_dp:=rng_to_dp;p_brg_to_dp:=brg_to_dp;trg_acq_rb:=FALSE;{Make sure Acquire is not currently running}if curgolstatus[3]<>1 then REF_IN_RNG, if ref_in_rng==TRUE & toi_found==FALSE then trg_acq_rb:=TRUE;				
Control Aircraft / Maintain SA	Monitor TSD	END				Prob.	0.05	Monitor A/C Systems Status
							0.95	Dummy 1: Workload Mgt

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Control Aircraft / Maintain SA	Monitor A/C Systems Status	START						
Control Aircraft / Maintain SA	Monitor A/C Systems Status	Check Fuel			p_fuel_qty:=ac_fuel_qty;			
Control Aircraft / Maintain SA	Monitor A/C Systems Status	END						
Control Aircraft / Maintain SA	Monitor Audio	START						
Control Aircraft / Maintain SA	Monitor Audio	Monitor Audio Channel		p_launch_ton:=launch_ton;p_rpln_ton:=rpln_ton;p_tgt_update:=tgt_update;p_rpln_avail:=rpln_avail;p_rpln_reasn:=rpln_reason;{following checks for presence of replan and sets nav trigger to TRUE if appropriate}if p_rpln_avail==TRUE & (rpln_reason==K_RP_TYP_THT |rpln_reason==K_RP_TYP_NAV) then trg_nav_pln:=TRUE;	if p_launch_ton==TRUE then CLR_ACQ_TRG, CLR_ATK_TRG, CLR_NAV_TRG, trg_evade:=TRUE;	Tactical	1-(p_tgt_update==TRUE & recording==FALSE & tgt_recorded==FALSE & toi_found==FALSE & check_upd==FALSE & halt_done==FALSE)	Monitor Audio Channel
							p_tgt_update==TRUE & recording==FALSE & tgt_recorded==FALSE & toi_found==FALSE & check_upd==FALSE & halt_done==FALSE	Dummy 1: Check for Tgt Update
							halt_done==TRUE	End

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Control Aircraft / Maintain SA	Monitor Audio	Receive/Copy Udated Target Coordinates	(WKLD_ACQ==FALSE & WKLD_NAV==FALSE & WKLD_ATK==FALSE & WKLD_FTP==FALSE & curgolstatus[1]<>1) ;	WKLD_FTP:=TRUE;recording:=TRUE;	tgt_recorded:=TRUE;recording:=FALSE;{the following triggers Acquisition, once update is recorded}trg_acq_upd:=TRUE;			
Control Aircraft / Maintain SA	Monitor Audio	Dummy 1: Check for Tgt Update		check_upd:=TRUE;		Multiple		Receive/Copy Udated Target Coordinates
								Monitor Audio Channel
Control Aircraft / Maintain SA	Monitor Audio	Dummy 2 - Handle Shotdown	(shotdown==TRUE) ;					
Control Aircraft / Maintain SA	Monitor Audio	Dummy 3: Workload Mgt		WKLD_FTP:=FALSE;				
Control Aircraft / Maintain SA	Monitor Audio	END						
Control Aircraft / Maintain SA	Initialize Vars	START						
Control Aircraft / Maintain SA	Initialize Vars	Dummy: Initialize Variables	(route>0) ;	{Once FRED sends us a route number, we can initialize variables}INIT_CMD_IDS;INIT_GLOBAL;INIT_ENTTYPE;INIT_SPECIFC;INIT_ALT_ACQ;if route==1 then INIT_LK_PT1;if route==2 then INIT_LK_PT2;if route==3 then INIT_LK_PT3;if route==4 then INIT_LK_PT4;if route==5 then INIT_LK_PT5;if route==6 then INIT_LK_PT6;INIT_SAR_LKQ;INIT_TIR_LKQ;				
Control Aircraft / Maintain SA	Initialize Vars	Dummy: Wait For Start		route:=0;		Tactical	if route > 0	Dummy: Initialize Variables
							if route==0	Dummy: Wait For Start
Control Aircraft / Maintain SA	Initialize Vars	Dummy for Testing Purposes		{hpm_des_thtl:=3;hpm_cmd_id:=19;hpm_rti_trig+=1;}1;				

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Control Aircraft / Maintain SA	Initialize Vars	END				Multiple		Monitor Flight Instruments
								Monitor Audio
Control Aircraft / Maintain SA		END						
Control Aircraft / Maintain SA		Dummy 2: Determine whether to end		if clock>start_timer+K_DELAY_END then hpm_cmd_id:=K_GAMEOVER, hpm_rti_trig+=1;	if clock>start_timer+K_DELAY_END then{halt or start end task}halt_done:=TRUE ;	Tactical	halt_done==FALSE	Monitor Flight Instruments
							halt_done==TRUE	Dummy 4: Rejoin
Control Aircraft / Maintain SA		Dummy 3: Check for Trigger Conditions		{this checks for toi_found in order to trigger trg_nav_fnd, and for atk_rpln_don in order to trigger trg_attack, it allows acquisition and navigation, respectively, to end normally}if end_acq==TRUE & toi_found==TRUE & atk_rpln_don==FALSE & (p_rng_to_dp>=K_LAR_RNG | absolute(p_brg_to_dp>30)) then trg_nav_fnd:=TRUE;if attack_done==FALSE & end_acq==TRUE & toi_found==TRUE & p_rng_to_dp<K_LAR_RNG & absolute(p_brg_to_dp)<=30 then {set atk_rpln_don to TRUE since the replan won't be needed} atk_rpln_don:=TRUE;if atk_rpln_don==TRUE & attack_done==FALSE & nav_fnd_pdg==FALSE then {we want to go into attack because we've found the target and we're in LAR rng and not in the process of replanning to attack} trg_attack:=TRUE;if attack_done==TRUE & abort_done==FALSE then trg_nav_abrt:=TRUE;				
Control Aircraft / Maintain SA		Dummy 1: Workload Mgt		WKLD_FTP:=FALSE;				
Control Aircraft / Maintain SA		Dummy 4: Rejoin						

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Evade	Monitor Audio	START						
Evade	Monitor Audio	Monitor Audio Channel			p_launch_ton:=launch_ton;p_rpln_ton:=rpln_ton;p_tgt_update:=tgt_update;p_rpln_avail:=rpln_avail;p_rpln_reasn:=rpln_reason;{following checks for presence of replan and sets nav trigger to TRUE if appropriate}if p_rpln_avail==TRUE & (rpln_reason==K_RP_TYP_THT |rpln_reason==K_RP_TYP_NAV) then trg_nav_pln:=TRUE;	Tactical	end_audio==FALSE & shotdown==FALSE;	Monitor Audio Channel
							end_audio==TRUE & shotdown==FALSE;	Dummy 2: Set audio_done
							shotdown==TRUE ;	Dummy 3: Handle Shotdown
Evade	Monitor Audio	Dummy 1: Init end_audio						
Evade	Monitor Audio	Dummy 2: Set audio_done		BEGEFFECT;	audio_done:=TRUE;			
Evade	Monitor Audio	Dummy 3: Handle Shotdown						
Evade	Monitor Audio	END						

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Evade	Execute Evasion Strategy	START						
Evade	Execute Evasion Strategy	Initiate Evade Maneuver		hpm_cmd_id:=K_EVADE_MNVR;hpm_rti_trig+=1;end_turn:=FALSE;end_evd_wkld:=FALSE;		Multiple		Maintain Evade Maneuver
								Check Heading
Evade	Execute Evasion Strategy	Maintain Evade Maneuver		end_turn:=FALSE;p_tht_count:=tht_count;	{previous command to initiate starts the manueuver. FRED will end the maneuver automatically when desired heading is achieved. This function simply maintains workload until that time}temp_hdg:=absolute(p_heading-hpm_des_hdg);if (temp_hdg<=K_HDG_ADJ | p_tht_count==0) then end_turn:=TRUE;	Tactical	end_turn==FALSE;	Maintain Evade Maneuver
							end_turn==TRUE & p_tht_count<>0;	Dummy: Rejoin prior to end
							end_turn==TRUE & p_tht_count==0	Abort Evade Maneuver
Evade	Execute Evasion Strategy	Release Chaff/Flare			hpm_cmd_id:=K_REL_CMEAS;hpm_rti_trig+=1;			
Evade	Execute Evasion Strategy	Retract TIR			hpm_cmd_id:=K_STOW_TIR;hpm_rti_trig+=1;tir_deployed:=FALSE;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Evade	Execute Evasion Strategy	Check Countermeasure Stores			p_chf_avail:=chf_avail; p_flr_avail:=flr_avail;	Tactical	1-(p_chf_avail>0 | p_flr_avail>0)	Initiate Evade Maneuver
							p_chf_avail>0 | p_flr_avail>0	Release Chaff/Flare
Evade	Execute Evasion Strategy	Dummy: Check Autopilot		BEGEFFECT;		Tactical	auto_pilot==ON	Dummy: Disengage Autopilot
							auto_pilot==OFF	Dummy: Check Airspeed
Evade	Execute Evasion Strategy	Dummy: Disengage Autopilot		BEGEFFECT;	hpm_cmd_id:=K_A_PILOT_D;hpm_rti_trig+=1;auto_pilot:=OFF;			
Evade	Execute Evasion Strategy	Dummy: Check Airspeed		BEGEFFECT;	p_airspeed:=ac_airspeed;p_power:=ac_power;	Tactical	cur_throttle==hpm_des_thtl	Check Countermeasure Stores
							cur_throttle<>hpm_des_thtl	Move Throttle
Evade	Execute Evasion Strategy	Move Throttle			hpm_cmd_id:=K_THROTTLE;hpm_rti_trig+=1;cur_throttle:=hpm_des_thtl;			
Evade	Execute Evasion Strategy	Check TIR				Tactical	tir_deployed==TRUE	Retract TIR
							tir_deployed==FALSE	Dummy: Check Autopilot

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Evade	Execute Evasion Strategy	Check Heading			p_heading:=ac_heading ;if end_turn==TRUE then end_evd_wkld:=TRUE;	Tactical	end_evd_wkld==FALSE	Check Attitude
							end_evd_wkld==TRUE	Dummy 1: End Instrument Wkld
Evade	Execute Evasion Strategy	Check Attitude			C_r:=random();if C_r>=0.000000 & C_r<0.800000 then p184:=1;if C_r>=0.800000 & C_r<1.000000 then p184:=2;p_pitch:=ac_pitch;p_roll:=ac_roll;	Prob.	0.8	Check Heading
							0.2	Check Altitude
Evade	Execute Evasion Strategy	Check Altitude			p_altitude:=ac_altitude;			
Evade	Execute Evasion Strategy	Dummy: Rejoin prior to end	(end_evd_wkld==TRUE & end_turn==TRUE) ;					
Evade	Execute Evasion Strategy	Abort Evade Maneuver		{command rollout at current heading}hpm_des_hdg:=p_heading;				
Evade	Execute Evasion Strategy	Dummy 1: End Instrument Wkld		BEGEFFECT;				
Evade	Execute Evasion Strategy	END						

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Evade	Select Evasion Strategy	START				Tactical	curr_emitter<>prev_emitter	Determine Maneuver to Intercept Minimization Path
							curr_emitter==prev_emitter	Determine Jink
Evade	Select Evasion Strategy	Determine Maneuver to Intercept Minimization Path		int_min_hdg:=p_heading+p_tht_emtbrg[top_threat]+180;if int_min_hdg>=360 then int_min_hdg-=360;if int_min_hdg<0 then int_min_hdg+=360;SET_1ST_JINK;	hpm_des_hdg:=int_min_hdg;hpm_des_thtl:=AFTERBURNER;prev_emitter:=curr_emitter;			
Evade	Select Evasion Strategy	Determine Jink		if next_jink==K_JINK_LEFT then hpm_des_hdg:=int_min_hdg-45, next_jink:=K_JINK_RIGHTelse hpm_des_hdg:=int_min_hdg+45, next_jink:=K_JINK_LEFT;	if hpm_des_hdg>=360 then hpm_des_hdg-=360;if hpm_des_hdg<0 then hpm_des_hdg+=360;			
Evade	Select Evasion Strategy	END						
Evade	Evaluate Threat Situation	START						
Evade	Evaluate Threat Situation	Identify Current Threat(s)		{ending effect gets type, range, bearing and time to intercept of all active threats (launched missiles)}p_tht_count:=tht_count;WKLD_NAV:=FALSE;WKLD_ACQ:=FALSE;WKLD_ATK:=FALSE;	tht_index:=0;while tht_index<tht_count do p_tht_emttyp[tht_index]:=tht_emttyp[tht_index], p_tht_emtrng[tht_index]:=tht_emtrng[tht_index], p_tht_emtbrg[tht_index]:=tht_emtbrg[tht_index], p_tht_msltti[tht_index]:=tht_msltti[tht_index], tht_index+=1;if p_tht_count==0 then must_evade:=FALSE;	Tactical	p_tht_count>0	Prioritize Threats
							p_tht_count==0	End

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Evade	Evaluate Threat Situation	Prioritize Threats		{time est from IMM. = 0.31 sec * (p1*(p1-1)/2) prioritization time.time is a calculated value using tht_count as p1 in the equation above.}p_heading:=ac_heading;{Call function to determine which threats are 'bad' (lethal) at Altitude}BAD_THREATS;{Call function to prioritize bad threats and set top_threat = to the most lethal (based on time to intercept)}RANK_THREATS;				
Evade	Evaluate Threat Situation	END						
Evade	Re-Engage Autopilot	START						
Evade	Re-Engage Autopilot	Engage Autopilot			hpm_cmd_id:=K_A_PILOT_E;hpm_rti_trig+=1;auto_pilot:=ON;			
Evade	Re-Engage Autopilot	Check Autopilot			if auto_pilot==ON then end_audio:=TRUE;	Tactical	auto_pilot==OFF	Engage Autopilot
							auto_pilot==ON	End
Evade	Re-Engage Autopilot	Engage Autothrottle		hpm_cmd_id:=K_A_THROTTLE;hpm_rti_trig+=1;	{set variable to end Monitor Audio}end_audio:=TRUE;			
Evade	Re-Engage Autopilot	Move Throttle		hpm_des_thtl:=CRUISE;hpm_cmd_id:=K_THROTTLE;hpm_rti_trig+=1;cur_throttle:=hpm_des_thtl;				
Evade	Re-Engage Autopilot	END						
Evade		START				Multiple		Evaluate Threat Situation
								Monitor Audio
Evade		Dummy 1: Choose whether to evade				Tactical	must_evade==TRUE	Select Evasion Strategy
							must_evade==FALSE	Re-Engage Autopilot

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Evade		Dummy 2: CLR_EVD_TRG	(end_audio==TRUE & audio_done==TRUE) ;	CLR_EVD_TRG;prev_emitter:=0;				
Evade		END						
Navigate	Plan For Reflying Target Area	START						
Navigate	Plan For Reflying Target Area	Put Cursor on Alt Point			hpm_curs_lat:=K_ALT_PT_LAT[route,alt_pt_index];hpm_curs_lon:=K_ALT_PT_LON[route,alt_pt_index];hpm_cmd_id:=K_MOVECURSOR;hpm_rti_trig+=1;alt_pt_index+=1;			
Navigate	Plan For Reflying Target Area	Designate as Mustfly Point			hpm_cmd_id:=K_DESIG_PT;hpm_rti_trig+=1;	Tactical	alt_pt_index<=K_NUM_ALT_PT[route];	Put Cursor on Alt Point
							alt_pt_index==(K_NUM_ALT_PT[route]+1);	Put Cursor on Release Point
							alt_pt_index>(K_NUM_ALT_PT[route]+1);	Select PLAN on UFC Planner Page
Navigate	Plan For Reflying Target Area	Put Cursor on Release Point			{make weapon release waypoint the current best guess of where the target is}{Store location of second (updated) wrp}upd_wrp_lat:=hpm_ref_lat;upd_wrp_lon:=hpm_ref_lon;hpm_curs_lat:=upd_wrp_lat;hpm_curs_lon:=upd_wrp_lon;hpm_cmd_id:=K_MOVECURSOR;hpm_rti_trig+=1;alt_pt_index+=1;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Navigate	Plan For Reflying Target Area	Select PLAN on UFC Planner Page			hpm_cmd_id:=K_REQ_RPLN;hpm_rti_trig+=1;{the following is for test purposes}p_rpln_avail:=TRUE;			
Navigate	Plan For Reflying Target Area	Evaluate Replan	((rpln_avail==TRUE)) ;	p_rpln_avail:=rpln_avail;p_rpln_reasn:=rpln_reason;		Tactical	p_rpln_reasn<>K_RP_TYP_ERR	Accept Replan
							p_rpln_reasn<>K_RP_TYP_ERR	End
Navigate	Plan For Reflying Target Area	Accept Replan			hpm_cmd_id:=K_U_ACP_RPLN;hpm_rti_trig+=1;pass:=2;{advance to next step in replanning (used in CHOOSE_DEST macro)}rp_next_step:=4;wpt_tmpclock:=clock;nav_nfnd_pdg:=FALSE;			
Navigate	Plan For Reflying Target Area	END						
Navigate	Plan For Attack	START						
Navigate	Plan For Attack	Put Cursor Over Target Icon			{move cursor to target of interest position}{Adjust index to compensate for HPM using 0 - n-1}index:=il_mover_idx[toi_index]-1;if index >= 0 & index < 30 then hpm_curs_lat:=mp_lat[index], hpm_curs_lon:=mp_lon[index], hpm_cmd_id:=K_MOVECURSOR, hpm_rti_trig+=1else {should log error}1;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Navigate	Plan For Attack	Designate as Mustfly Point			hpm_cmd_id:=K_DESIG_PT;hpm_rti_trig+=1;			
Navigate	Plan For Attack	Select PLAN on UFC Planner Page			hpm_cmd_id:=K_REQ_RPLN;hpm_rti_trig+=1;{the following is for test purposes}p_rpln_avail:=TRUE;			
Navigate	Plan For Attack	Evaluate Replan	((rpln_avail==TRUE)) ;	p_rpln_avail:=rpln_avail;p_rpln_reasn:=rpln_reason;		Tactical	p_rpln_reasn<>K_RP_TYP_ERR	Accept Replan
							p_rpln_reasn==K_RP_TYP_ERR	End
Navigate	Plan For Attack	Accept Replan			hpm_cmd_id:=K_U_ACP_RPLN;hpm_rti_trig+=1;atk_rpln_don:=TRUE;nav_fnd_pdg:=FALSE;			
Navigate	Plan For Attack	END						
Navigate	Respond to Auto Replan	START						
Navigate	Respond to Auto Replan	Evaluate Replan	(WKLD_FTP==FALSE) ;	WKLD_NAV:=TRUE;				
Navigate	Respond to Auto Replan	Accept Replan			hpm_cmd_id:=K_F_ACP_RPLN;hpm_rti_trig+=1;p_rpln_avail:=FALSE;			
Navigate	Respond to Auto Replan	Check DOI		p_curr_doi:=current_doi;		Tactical	p_curr_doi==K_DOI_CENTER	Accept Replan
							p_curr_doi<>K_DOI_CENTER	Select TSD as DOI

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Navigate	Respond to Auto Replan	Select TSD as DOI			hpm_cmd_id:=K_DOI_CTRMPD;hpm_rti_trig+=1;current_doi:=K_DOI_CENTER;p_curr_doi:=current_doi;			
Navigate	Respond to Auto Replan	END						
Navigate	Abort	START						
Navigate	Abort	Select ABRT on UFC Planner Page			hpm_cmd_id:=K_REQ_ABORT;hpm_rti_trig+=1;{Wait up to 15 sec for replan to complete}replan_timer:=clock+(15/60);			
Navigate	Abort	Evaluate Replan		p_rpln_avail:=rpln_avail;p_rpln_reasn:=rpln_reason;		Tactical	p_rpln_reasn<>K_RP_TYP_ERR	Accept Replan
							p_rpln_reasn==K_RP_TYP_ERR	Dummy 2: Start End Timers
Navigate	Abort	Accept Replan			hpm_cmd_id:=K_U_ACP_RPLN;hpm_rti_trig+=1;{capture clock time, used in fly the plane to determine whether to end the trial}start_timer:=clock;abort_done:=TRUE;{set rp_next_step to 99 such that CHOOSE_DEST macro will not fire again}rp_next_step:=99;{clear other triggers for pending requested replans}nav_nfnd_pdg:=FALSE;nav_fnd_pdg:=FALSE;nav_abrt_pdg:=FALSE;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Navigate	Abort	Dummy 2: Start End Timers		{capture clock time, used in fly the plane to determine whether to end the trial}start_timer:=clock;abort_done:=TRUE;{set rp_next_step to 99 such that CHOOSE_DEST macro will not fire again}rp_next_step:=99;{clear other triggers for pending requested replans}nav_nfnd_pdg:=FALSE;nav_fnd_pdg:=FALSE;nav_abrt_pdg:=FALSE;				
Navigate	Abort	Dummy 1: Check Replan Timer				Tactical	rpln_avail==TRUE	Evaluate Replan
							rpln_avail==FALSE & clock>replan_timer	Dummy 2: Start End Timers
							rpln_avail==FALSE & clock<=replan_timer	Dummy 1: Check Replan Timer
Navigate	Abort	END						
Navigate	Select Planner Mode and DOI	START						
Navigate	Select Planner Mode and DOI	Select Planner Mode on UFC			hpm_cmd_id:=K_UFC_PLNMOD;hpm_rti_trig+=1;ufc_mode:=K_UFC_PLAN;p_ufc_mode:=ufc_mode;			
Navigate	Select Planner Mode and DOI	Check UFC Mode	(WKLD_FTP==FALSE) ;		p_ufc_mode:=ufc_mode;	Tactical	p_ufc_mode<>K_UFC_PLAN	Select Planner Mode on UFC
							p_ufc_mode==K_UFC_PLAN	Check DOI
Navigate	Select Planner Mode and DOI	Check DOI			p_curr_doi:=current_doi;	Tactical	p_curr_doi<>K_DOI_CENTER	Select TSD as DOI
							p_curr_doi==K_DOI_CENTER	End

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Navigate	Select Planner Mode and DOI	Select TSD as DOI			hpm_cmd_id:=K_DOI_CTRMPD;hpm_rti_trig+=1;current_doi:=K_DOI_CENTER;p_curr_doi:=current_doi;			
Navigate	Select Planner Mode and DOI	END						
Navigate		START						
Navigate		Dummy 1: Set Replan Type		WKLD_ACQ:=FALSE;WKLD_ATK:=FALSE;	{set variables indicating that a requested replan is pending, such that if halted by evade, once evasion is done, pilot will restart the requested replan}if trg_nav_nfnd==TRUE then nav_nfnd_pdg:=TRUE; if trg_nav_fnd==TRUE then nav_fnd_pdg:=TRUE;if trg_nav_abrt==TRUE then nav_abrt_pdg:=TRUE;	Tactical	1-(trg_nav_fnd==TRUE | trg_nav_abrt==TRUE | trg_nav_nfnd==TRUE | nav_nfnd_pdg==TRUE)	Respond to Auto Replan
							trg_nav_fnd==TRUE | trg_nav_abrt==TRUE | trg_nav_nfnd==TRUE | nav_nfnd_pdg==TRUE	Select Planner Mode and DOI

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Navigate		Dummy 2: Set Requested Replan Type			{initialize alt_pt_index}alt_pt_index:=1;	Tactical	trg_nav_nfnd==TRUE | nav_nfnd_pdg==TRUE	Plan For Reflying Target Area
							trg_nav_fnd==TRUE | nav_fnd_pdg==TRUE	Plan For Attack
							trg_nav_abrt==TRUE | nav_abrt_pdg==TRUE	Abort
Navigate		Dummy 3: Trigger Reset		CLR_NAV_TRG;				
Navigate		Dummy 4: Handle Replanner Error				Tactical	rpln_reason==K_RP_TYP_ERR	Abort
							rpln_reason<>K_RP_TYP_ERR	Dummy 9: Workload Mgt
Navigate		Dummy 5: Workload Mgt		{duration is 0.5 sec to ensure that replan available gets set back to TRUE. This precludes the possibility of Triggering Nav twice for the same replan}1;	WKLD_NAV:=FALSE;			
Navigate		Dummy 9: Workload Mgt		WKLD_NAV:=FALSE;				
Navigate		END						
Acquire	Evaluate Image	START						

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Evaluate Image	Detect Objects	(il_image_id <> prev_img_id & WKLD_FTP==FALSE & curgolstatus[2]<>1 & curgolstatus[1]<>1) ;	WKLD_ACQ:=TRUE;{process image list to determine detected objects}{This produces data to be used in the iar* structure}DETECT_OBJ;				
Acquire	Evaluate Image	Identify Objects		{process image list to determine identified objects}{This populates the iar* structure}IDENTIFY_OBJ;if toi_found==TRUE then{set rp_next_step to 99 such that CHOOSE_DEST macro will not fire again} rp_next_step:=99;CHK_ID_2_IMG;{Send the Image Analysis Results to FRED}iar_rti_trig+=1;	{Tell FRED to process the previously sent IAR structure}hpm_cmd_id:=K_PROCES_OOI;hpm_rti_trig+=1;{reset alt_pt_inuse}alt_pt_inuse:=FALSE;if lat_to_image==hpm_ref_lat & lon_to_image==hpm_ref_lon then need_reflook:=FALSE;if pre_tir_pndg==TRUE then pre_tir_pndg:=FALSE, pre_tir_upd:=TRUE;			
Acquire	Evaluate Image	END						
Acquire	Update Shootlist	START						
Acquire	Update Shootlist	Check DOI			p_curr_doi:=current_doi;if sensor_2_use<=K_NSAR_HIGH then next_doi:=K_DOI_LEFTelse next_doi:=K_DOI_RIGHT;	Tactical	del_sl_count>0 {something to delete} & current_doi<>next_doi;	Check NTS
							del_sl_count>0 {something to delete} & current_doi<>next_doi;	Select DOI
							del_sl_count<=0 {nothing to delete} & add_sl_count>0 {something to add};	Place Cursor on Object to Add

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Update Shootlist	Check NTS			C_r:=random();if C_r>=0.000000 & C_r<0.330000 then p266:=1;if C_r>=0.330000 & C_r<1.000000 then p266:=2;	Prob.	0.3333	Bump NTS
							0.6666	Undesignate to Remove from Shootlist
Acquire	Update Shootlist	Bump NTS						
Acquire	Update Shootlist	Select DOI			if sensor_2_use<=K_NSAR_HIGH then hpm_cmd_id:=K_DOI_LFTMPD, hpm_rti_trig+=1, current_doi:=K_DOI_LEFT;if sensor_2_use>K_NSAR_HIGH then hpm_cmd_id:=K_DOI_RTMPD, hpm_rti_trig+=1, current_doi:=K_DOI_RIGHT;p_curr_doi:=current_doi;	Tactical	pl_num_del>0 {something to delete}	Check NTS
Acquire	Update Shootlist	Undesignate to Remove from Shootlist		del_sl_count-=1;		Tactical	pl_num_del<=0 {nothing to delete}	Place Cursor on Object to Add
							del_sl_count <= 0 & add_sl_count > 0	Check DOI
							del_sl_count > 0	Check NTS
							del_sl_count <= 0 & add_sl_count <= 0	End
Acquire	Update Shootlist	Place Cursor on Object to Add						

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Update Shootlist	Designate to Add Detection to Shootlist		add_sl_count-=1;		Tactical	add_sl_count>0	Place Cursor on Object to Add
							add_sl_count==0	End
Acquire	Update Shootlist	Dummy 1: Set Number of Items to Add/Del		{create local counter that will be decremented each time an add/del is made}del_sl_count:=pl_num_del;add_sl_count:=pl_num_add ;				
Acquire	Update Shootlist	END						
Acquire	Designate Target	START						
Acquire	Designate Target	Check DOI			p_curr_doi:=current_doi;	Tactical	current_doi<>K_DOI_LEFT & sensor_2_use<=K_NSAR_HIGH	Select Radar As DOI
							(current_doi==K_DOI_LEFT & sensor_2_use<=K_NSAR_HIGH) | (current_doi==K_DOI_RIGHT & sensor_2_use>K_NSAR_HIGH)	Slew Cursor to Target Aimpoint
							current_doi<>K_DOI_RIGHT & sensor_2_use>K_NSAR_HIGH	Select TIR as DOI
Acquire	Designate Target	Select Radar As DOI			hpm_cmd_id:=K_DOI_LFTMPD;hpm_rti_trig+=1;current_doi:=K_DOI_LEFT;p_curr_doi:=current_doi;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Designate Target	Designate as Next to Shoot			hpm_cmd_id:=K_DESIG_PT;hpm_rti_trig+=1; end_acq:=TRUE;			
Acquire	Designate Target	Slew Cursor to Target Aimpoint			{move cursor to current target of interest position}index:=il_mover_idx[toi_index]-1;if index >= 0 & index < 30 then hpm_curs_lat:=mp_lat[index], hpm_curs_lon:=mp_lon[index], hpm_cmd_id:=K_MOVECURSOR, hpm_rti_trig+=1else {should log error here}1;			
Acquire	Designate Target	Select TIR as DOI			hpm_cmd_id:=K_DOI_RTMPD;hpm_rti_trig+=1;current_doi:=K_DOI_RIGHT;p_curr_doi:=current_doi;			
Acquire	Designate Target	END						
Acquire	Update Tgt Location	START						
Acquire	Update Tgt Location	Check UFC Mode			p_ufc_mode:=ufc_mode;	Tactical	p_ufc_mode<>K_UFC_TGT	Select Target Mode on UFC
							p_ufc_mode==K_UFC_TGT	Enter Updated Tgt Lat/Lon in UFC
Acquire	Update Tgt Location	Select Target Mode on UFC			hpm_cmd_id:=K_UFC_TGTMOD;hpm_rti_trig+=1;ufc_mode:=K_UFC_TGT;p_ufc_mode:=ufc_mode;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Update Tgt Location	Enter Updated Tgt Lat/Lon in UFC		{send updated target lat/lon back to FRED}hpm_upd_lat:=updated_lat;hpm_upd_lon:=updated_lon;hpm_cmd_id:=K_UFC_TGT_LL;hpm_rti_trig+=1;				
Acquire	Update Tgt Location	Accept Updated Tgt Lat/Lon on UFC		{accept the entered lat/lon}hpm_cmd_id:=K_UFC_LL_ACP;hpm_rti_trig+=1;	{tell FRED to use the updated lat/lon as the new target reference location}hpm_ref_lat:=updated_lat;hpm_ref_lon:=updated_lon;hpm_cmd_id:=K_UPD_REFLOC;hpm_rti_trig+=1;need_reflook:=TRUE;pre_tir_upd:=FALSE;pre_tir_pndg:=FALSE;tgt_upd_done:=TRUE;			
Acquire	Update Tgt Location	END						
Acquire	Decide Where/How to Look	START						
Acquire	Decide Where/How to Look	Estimate Range and Bearing to Ref Point	(WKLD_FTP==FALSE & curgolstatus[2]<>1 & curgolstatus[1]<>1) ;	WKLD_ACQ:=TRUE;{Init variable that we are not ready to end acquisition}{Will be set TRUE in Dummy7-Time to End Acquisition if reached}end_acq:=FALSE;alt_pt_inuse:=FALSE;	p_rng_to_ref:=rng_to_ref;p_brg_to_ref:=brg_to_ref;{call macro that will return ref_in_rng, in_tir_rng, & in_sar_rng}REF_IN_RNG;	Tactical	ref_in_rng==TRUE	Dummy 1: Have Fred Rank OOI
							ref_in_rng==FALSE	Dummy 7: Time to End Acquisition
Acquire	Decide Where/How to Look	Dummy 1: Have Fred Rank OOI		{Command Prioritize_OOI}lastpl_count:=pl_count;hpm_cmd_id:=K_RANK_OOI;hpm_rti_trig+=1;		Tactical	in_tir_rng==FALSE	Dummy 2: Handle SAR Look Points
							in_tir_rng==TRUE	Dummy 3: Handle TIR Conditions

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Decide Where/How to Look	Dummy 7: Time to End Acquisition			{no shootlist items are in rng/brg or shootlist is empty and no alt look points are in rng/brg.}end_acq:=TRUE;			
Acquire	Decide Where/How to Look	Dummy 6: Ready to Image						
Acquire	Decide Where/How to Look	Dummy 2: Handle SAR Look Points		{since we're getting close to TIR rng (within 3 mi), take a real beam look to add/update movers}if rng_to_ref<23 & pre_tir_upd==FALSE then pre_tir_pndg:=TRUE, lat_to_image:=hpm_ref_lat, lon_to_image:=hpm_ref_lon, sensor_2_use:=K_REAL_BEAMelse {get lat_to_image, lon_to_image, sensor_2_use for next point on look queue} SET_SAR_LKPT;id_to_image:=(-1);				
Acquire	Decide Where/How to Look	Dummy 3: Handle TIR Conditions	(pl_count<>lastpl_count) ;			Tactical	pl_number>1 & pl_viable==TRUE & need_reflook==FALSE;	Dummy 4: Handle the PL
							1-(pl_number>1 & pl_viable==TRUE & need_reflook==FALSE);	Dummy 5: Handle TIR Look Pts
Acquire	Decide Where/How to Look	Dummy 4: Handle the PL		{here, the priority list includes items in addition to the ref point. We're also in TIR range, which allows us to try to identify with TIR. So, set lat/lon to image = to top item on the priority list.}lat_to_image:=pl_lat[0];lon_to_image:=pl_lon[0];sensor_2_use:=pl_nxtsensor[0];{protect against zooming in on 'stale' point}if id_to_image<>pl_id[0] & sensor_2_use==K_TIR_2X_NAR then sensor_2_use:=K_TIR_NARROW;id_to_image:=pl_id[0];				

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Decide Where/How to Look	Dummy 5: Handle TIR Look Pts		{if we're still in sar range & gimbal limits, and there is nothing on the PL, continue using radar to take advantage of gmti}if in_sar_rng==TRUE then SET_SAR_LKPTelse SET_TIR_LKPT;id_to_image:=(-1);				
Acquire	Decide Where/How to Look	Choose Sensor and Location to Image		{look at sensor/location to determine decision duration}dec_time:=0;if lat_to_image==hpm_curs_lat & lon_to_image==hpm_curs_lon then {simple decision to zoom in} dec_time:=.07;if dec_time==0 & lat_to_image==hpm_ref_lat & lon_to_image==hpm_ref_lon then {simple decision check ref} dec_time:=.07;if dec_time==0 & lat_to_image==pl_lat[0] & lon_to_image==pl_lon[0] & pl_number==2 then {only one pl item to consider} dec_time:=.07;if dec_time==0 & lat_to_image==pl_lat[0] & lon_to_image==pl_lon[0] & pl_number<4 then {prioritization time from micromodel} dec_time:= ((pl_number-1)*((pl_number-1)-1)/2)*0.31;if dec_time==0 & lat_to_image==pl_lat[0] & lon_to_image==pl_lon[0] & pl_number>=4 then {prioritization time from micromodel, assuming pilot considers only 3 nearest points} dec_time:= (3*(3-1)/2)*0.31;if dec_time==0 then {Else map reading w/ 1 association attempt} dec_time:=7;				
Acquire	Decide Where/How to Look	END						
Acquire	Image Target	START						
Acquire	Image Target	Dummy 3: Move or Bump?		{tactical decision determines whether we slew to an alt point, bump to a new pl item or zoom in on a pl item. the following code checks to see if we need a slight slew to center a mover before zooming in} slew_needed:=FALSE;{check to see if we're zooming in on a previously detected object, and that object is no longer centered under the cursor. If so, we need to slew the cursor}if zoom_only==TRUE & (lat_to_image<>hpm_curs_lat | lon_to_image<>hpm_curs_lon) then slew_needed:=TRUE;		Tactical	alt_pt_inuse==TRUE;	Slew Cursor to Alt Look Location
							alt_pt_inuse==FALSE & zoom_only==FALSE;	Bump NTS

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
							alt_pt_inuse==FALSE & zoom_only==TRUE & slew_needed==FALSE;	Select FOV/Command Image
							alt_pt_inuse==FALSE & zoom_only==TRUE & slew_needed==TRUE;	Slew Cursor to Object
Acquire	Image Target	Dummy 4: Need to Designate?				Tactical	sensor_2_use<=K_NSAR_HIGH;	Designate as Next to Shoot
							Check DOI for Imaging	Check DOI for Imaging
Acquire	Image Target	Check TIR				Tactical	sensor_2_use>K_NSAR_HIGH & tir_deployed==FALSE;	Deploy TIR
							sensor_2_use<=K_NSAR_HIGH & tir_deployed==TRUE;	Retract TIR
							1;	Dummy 1: Check Zoom Condition & Initial DOI Req's
Acquire	Image Target	Deploy TIR			hpm_cmd_id:=K_DEPLOY_TIR;hpm_rti_trig+=1;tir_deployed:=TRUE;{set default TIR FOV to wide when first deployed}tir_fov:=K_TIR_WIDE;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Image Target	Retract TIR			hpm_cmd_id:=K_STOW_TIR;hpm_rti_trig+=1;tir_deployed:=FALSE;			
Acquire	Image Target	Dummy 1: Check Zoom Condition & Initial DOI Req'		{initialize 'need_planchk', which is a flag to see if pilot needs to change into planner mode before adding new lookpoints}need_planchk:=FALSE;{determine initial DOI}if sensor_2_use<=K_NSAR_HIGH & alt_pt_inuse==FALSE then next_doi:=K_DOI_LEFT;if sensor_2_use<=K_NSAR_HIGH & alt_pt_inuse==TRUE then next_doi:=K_DOI_CENTER, need_planchk:=TRUE;if sensor_2_use>K_NSAR_HIGH then next_doi:=K_DOI_RIGHT;	{determine whether next look is simply a zoom in on previous look}zoom_only:=FALSE;if last_sensor < K_TIR_WIDE then last_snsr_cl:=1else last_snsr_cl:=2;if sensor_2_use < K_TIR_WIDE then cmd_snsr_cl:=1else cmd_snsr_cl:=2;{zoom only if sensor type is the same, doi is current, and cursor is already in desired position}if lat_to_image==hpm_curs_lat & lon_to_image==hpm_curs_lon & last_snsr_cl==cmd_snsr_cl & current_doi==next_doi then zoom_only:=TRUE;	Tactical	zoom_only==TRUE	Dummy 3: Move or Bump?
							zoom_only==FALSE	Check DOI for Designation

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Image Target	Check DOI for Designation			p_curr_doi:=current_doi;	Tactical	current_doi==next_doi & need_planchk==FALSE	Dummy 3: Move or Bump?
							current_doi<>next_doi	Select DOI for Designation
							current_doi==next_doi & need_planchk==TRUE	Check UFC Mode
Acquire	Image Target	Select DOI for Designation		if next_doi==K_DOI_LEFT then hpm_cmd_id:=K_DOI_LFTMPD;if next_doi==K_DOI_CENTER then hpm_cmd_id:=K_DOI_CTRMPD;if next_doi==K_DOI_RIGHT then hpm_cmd_id:=K_DOI_RTMPD;hpm_rti_trig+=1;current_doi:=next_doi;p_curr_doi:=current_doi;		Tactical	need_planchk==FALSE	Dummy 3: Move or Bump?
							need_planchk==TRUE	Check UFC Mode
Acquire	Image Target	Slew Cursor to Alt Look Location		{time est from observation: =2 s}1;	{command FRED to move the cursor to next Alt Look Point}hpm_curs_lat:=lat_to_image;hpm_curs_lon:=lon_to_image;hpm_cmd_id:=K_MOVECURSOR;hpm_rti_trig+=1;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Image Target	Bump NTS			{NOTE: Due to the fact that the FRED Shootlist is not fully utilized, we move the cursor to the appropriate location and assign time as if pilot had to bump through the shootlist}hpm_curs_lat:=lat_to_image;hpm_curs_lon:=lon_to_image;hpm_cmd_id:=K_MOVECURSOR;hpm_rti_trig+=1;			
Acquire	Image Target	Designate as Next to Shoot			hpm_cmd_id:=K_DESIG_PT;hpm_rti_trig+=1;			
Acquire	Image Target	Select FOV/Command Image		{if sensor is TIR, select fov from stick, therefore use switch hit}{time est from IMM: ((.4 push button/toggle}{if sensor is radar, select fov from display. Therefore use Fitts law for time}{time est from IMM: 1 sec hand movement (Fitts Law - Welford variant, 66 cm dist and 2cm tgt) + .4 push button + .3 fixation + .1 percept process= 1.8 sec}1;if sensor_2_use>K_NSAR_HIGH then task_time:=.4else task_time:=1.8;	if sensor_2_use>K_NSAR_HIGH then last_sensor:=sensor_2_use, prev_img_id:=il_image_id, hpm_cmd_id:=K_GEN_IMAGE, hpm_rti_trig+=1;{else, we command sar in following dummy task to allow for time after request needed to build image}1;			
Acquire	Image Target	Check UFC Mode			p_ufc_mode:=ufc_mode;	Tactical	p_ufc_mode==K_UFC_PLAN	Select Target Mode on UFC
							p_ufc_mode<>K_UFC_PLAN	Dummy 2: Clear SL

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Image Target	Select Target Mode on UFC			hpm_cmd_id:=K_UFC_TGTMOD;hpm_rti_trig+=1;ufc_mode:=K_UFC_TGT;p_ufc_mode:=ufc_mode;			
Acquire	Image Target	Dummy 2: Clear SL		{command FRED to Clear current shootlist item}hpm_cmd_id:=K_RMV_SL_OBJ;hpm_rti_trig+=1;				
Acquire	Image Target	Check DOI for Imaging			p_curr_doi:=current_doi;	Tactical	(p_curr_doi==K_DOI_LEFT & sensor_2_use<=K_NSAR_HIGH) | (p_curr_doi==K_DOI_RIGHT & sensor_2_use>K_NSAR_HIGH)	Select FOV/Command Image
							1-((p_curr_doi==K_DOI_LEFT & sensor_2_use<=K_NSAR_HIGH) | (p_curr_doi==K_DOI_RIGHT & sensor_2_use>K_NSAR_HIGH))	Select DOI for Imaging
Acquire	Image Target	Select DOI for Imaging			if sensor_2_use<=K_NSAR_HIGH then hpm_cmd_id:=K_DOI_LFTMPD, current_doi:=K_DOI_LEFTelse hpm_cmd_id:=K_DOI_RTMPD, current_doi:=K_DOI_RIGHT;hpm_rti_trig+=1; p_curr_doi:=current_doi;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire	Image Target	Slew Cursor to Object		{move cursor over the object of interest & re-assign lat/lon to image}lat_to_image:=pl_lat[0];lon_to_image:=pl_lon[0];hpm_curs_lat:=lat_to_image;hpm_curs_lon:=lon_to_image;hpm_cmd_id:=K_MOVECURSOR;hpm_rti_trig+=1;				
Acquire	Image Target	END						
Acquire		START						
Acquire		Dummy 1: Wkld Mgt and check for update	(WKLD_FTP==FALSE & curgolstatus[2]<>1 & curgolstatus[1]<>1) ;	WKLD_ATK:=FALSE;WKLD_ACQ:=TRUE;		Tactical	tgt_recorded==TRUE & tgt_upd_done<>TRUE	Update Tgt Location
							1-(tgt_recorded==TRUE & tgt_upd_done<>TRUE)	Decide Where/How to Look
Acquire		Dummy 2: Target Found?		{if target is identified in IDENTIFY_OBJ &CALC_IDENT macros, start the Designate Target Function and end the DECIDE, IMAGE, EVALUATE, UPDATE function loop}lastpl_count:=pl_count;hpm_cmd_id:=K_RANK_OOI;hpm_rti_trig+=1;		Tactical	toi_found==TRUE	Designate Target
							toi_found==FALSE	Dummy 4: Need to update PL?
Acquire		Dummy 3: End Acquisition				Tactical	end_acq==FALSE	Image Target
							end_acq==TRUE	Check TIR
Acquire		Dummy 7: Clear Acq Trigger		CLR_ACQ_TRG;				
Acquire		Check TIR				Tactical	tir_deployed==TRUE	Retract TIR
							tir_deployed==FALSE	Dummy 10: Workload Mgt
Acquire		Retract TIR			hpm_cmd_id:=K_STOW_TIR;hpm_rti_trig+=1;tir_deployed:=FALSE;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Acquire		Dummy 4: Need to update PL?	(lastpl_count<>pl_count) ;	{after shootlist has come back, tactical decision looks at pl_num_add and pl_num_del to determine whether an update is necessary. These were rec'd in the PL interaction}1;		Tactical	pl_num_add > 0 | pl_num_del > 0	Update Shootlist
							1-(pl_num_add > 0 | pl_num_del > 0)	Dummy 9: Workload Mgt
Acquire		Dummy 6: Workload Mgt		WKLD_ACQ:=FALSE;				
Acquire		Dummy 8: Workload Mgt		{allow Fly The Plane to resume if waiting for a radar image}if sensor_2_use<=K_NSAR_HIGH then WKLD_ACQ:=FALSE;		Tactical	sensor_2_use>K_NSAR_HIGH	Evaluate Image
							sensor_2_use<=K_NSAR_HIGH	Dummy 5: Wait for radar image
Acquire		Dummy 10: Workload Mgt		WKLD_ACQ:=FALSE;				
Acquire		Dummy 5: Wait for radar image		{allowing 6 sec for radar paint/ image gen}1;	{sensor_2_use, which indicates FOV, has already been set}last_sensor:=sensor_2_use;prev_img_id:=il_image_id;hpm_cmd_id:=K_GEN_IMAGE;hpm_rti_trig+=1;			
Acquire		Dummy 9: Workload Mgt		WKLD_ACQ:=FALSE;				
Acquire		END						
Attack	Monitor Target Position and Range	START						
Attack	Monitor Target Position and Range	Check LAR on TSD			p_rng_to_dp:=rng_to_dp;p_brg_to_dp:=brg_to_dp;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Attack	Monitor Target Position and Range	Monitor Tgt Position in TIR				Tactical	tir_trk_stat==1	Check LAR on TSD
							tir_trk_stat<>1	Slew TIR to Track Tgt
Attack	Monitor Target Position and Range	Slew TIR to Track Tgt			{move cursor to current target of interest position}index:=il_mover_idx[toi_index]-1;if index >= 0 & index < 30 then hpm_curs_lat:=mp_lat[index], hpm_curs_lon:=mp_lon[index], hpm_cmd_id:=K_MOVECURSOR, hpm_rti_trig+=1else {should log error here}1;			
Attack	Monitor Target Position and Range	Check DOI	(WKLD_FTP==FALSE & curgolstatus[1]<>1 &curgolstatus[2]<>1 & curgolstatus[3]<>1) ;	WKLD_ATK:=TRUE;	p_curr_doi:=current_doi;	Tactical	current_doi==K_DOI_RIGHT	Monitor Tgt Position in TIR
							current_doi<>K_DOI_RIGHT	Select TIR as DOI
Attack	Monitor Target Position and Range	Select TIR as DOI		hpm_cmd_id:=K_DOI_RTMPD;hpm_rti_trig+=1;current_doi:=K_DOI_RIGHT;p_curr_doi:=current_doi;				
Attack	Monitor Target Position and Range	Designate as Next to Shoot			hpm_cmd_id:=K_DESIG_PT;hpm_rti_trig+=1;			

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Attack	Monitor Target Position and Range	END						
Attack	Release Weapon	START						
Attack	Release Weapon	Command Weapon Release			hpm_cmd_id:=K_REL_WCM;hpm_rti_trig+=1;{start a timer that will have pilot track target for 15 sec after release, before stowing TIR}wpn_clock:=clock;			
Attack	Release Weapon	Slew Cursor to Target Aimpoint			{move cursor to current target of interest position}index:=il_mover_idx[toi_index]-1;if index >= 0 & index < 30 then hpm_curs_lat:=mp_lat[index], hpm_curs_lon:=mp_lon[index], hpm_cmd_id:=K_MOVECURSOR, hpm_rti_trig+=1else {should log error here}1;			
Attack	Release Weapon	Designate as Next to Shoot			hpm_cmd_id:=K_DESIG_PT;hpm_rti_trig+=1;			
Attack	Release Weapon	Check DOI	(WKLD_FTP==FALSE & curgolstatus[1]<>1 &curgolstatus[2]<>1 & curgolstatus[3]<>1) ;	WKLD_ATK:=TRUE;	p_curr_doi:=current_doi;	Tactical	p_curr_doi==K_DOI_RIGHT & tir_trk_stat==1	Command Weapon Release
							p_curr_doi==K_DOI_RIGHT & tir_trk_stat<>1	Slew Cursor to Target Aimpoint
							p_curr_doi<>K_DOI_RIGHT	Select TIR as DOI

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Attack	Release Weapon	Select TIR as DOI			hpm_cmd_id:=K_DOI_RTMPD;hpm_rti_trig+=1;current_doi:=K_DOI_RIGHT;p_curr_doi:=current_doi;	Tactical	tir_trk_stat==1	Command Weapon Release
							tir_trk_stat<>1	Slew Cursor to Target Aimpoint
Attack	Release Weapon	Retract TIR			if tir_deployed==TRUE then hpm_cmd_id:=K_STOW_TIR, hpm_rti_trig+=1, tir_deployed:=FALSE;attack_done:=TRUE;			
Attack	Release Weapon	Monitor Tgt Position in TIR				Tactical	(clock-wpn_clock)>.25;	Retract TIR
							tir_trk_stat<>1 & (clock-wpn_clock)<=.25;	Slew TIR to Track Tgt
							tir_trk_stat==1 & (clock-wpn_clock)<=.25;	Dummy: Delay Task
Attack	Release Weapon	Slew TIR to Track Tgt			{move cursor to current target of interest position}index:=il_mover_idx[toi_index]-1;if index >= 0 & index < 30 then hpm_curs_lat:=mp_lat[index], hpm_curs_lon:=mp_lon[index], hpm_cmd_id:=K_MOVECURSOR, hpm_rti_trig+=1else {should log error here}1;			
Attack	Release Weapon	Dummy: Delay Task						

Table F-1. Model Code Release Conditions, Beginning Effects, Ending Effects, and Decision Nodes (continued)

Goal	Function Name	Task Name	Release Condition	Beginning Effect	Ending Effect	Decision Type	Decision Logic	Task / Function Following Decision
Attack	Release Weapon	Re-Designate to Track			hpm_cmd_id:=K_DESIG_PT;hpm_rti_trig+=1;			
Attack	Release Weapon	END						
Attack		START						
Attack		Dummy 2: Trigger Reset		CLR_ATK_TRG;				
Attack		Dummy 1: Check for in Rng				Tactical	p_rng_to_dp>=K_LAR_RNG | absolute(p_brg_to_dp)>30	Monitor Target Position and Range
							p_rng_to_dp<K_LAR_RNG & absolute(p_brg_to_dp)<=30	Release Weapon
Attack		Check TIR				Tactical	tir_deployed==TRUE	Dummy 1: Check for in Rng
							tir_deployed==FALSE	Deploy TIR
Attack		Deploy TIR			hpm_cmd_id:=K_DEPLOY_TIR;hpm_rti_trig+=1;tir_deployed:=TRUE;			
Attack		Dummy 3: Workload Mgt		WKLD_ATK:=FALSE;				
Attack		Dummy 4: Workload Mgt		WKLD_ATK:=FALSE;				
Attack		Dummy 5: Check for in Range				Tactical	p_rng_to_dp<K_LAR_RNG & absolute(p_brg_to_dp)<=30	Release Weapon
							p_rng_to_dp>=K_LAR_RNG | absolute(p_brg_to_dp)>30	Dummy 3: Workload Mgt
Attack		END						

This page left blank intentionally

APPENDIX G

DESCRIPTIONS AND CODE ASSOCIATED WITH THE USER-DEFINED MACROS USED IN CASE STUDY 1

This page left blank intentionally

Table G-1. Descriptions and Code Associated with the User-Defined Macros Used in Case Study 1

<table>
<tr><th>Macro Name</th><th>Description</th><th>Macro Code</th></tr>
<tr><td>BAD_THREATS</td><td>In the scenario, there are two threats capable of reaching the aircraft at the planned altitude. This macro is used to check the emitter type of any launched threat and to determine whether that threat is one of the two lethal threat types.</td><td>{determine which of launched threats can kill you at altitude}
tht_index:=0;
must_evade:=FALSE;
while tht_index<tht_count do
if tht_emttyp[tht_index] == K_EMT_TYPE_1 |
tht_emttyp[tht_index] == K_EMT_TYPE_2 then
bad_tht[tht_index]:=TRUE,
must_evade:=TRUE,
tht_index+=1
else
bad_tht[tht_index]:=FALSE,
tht_index+=1;</td></tr>
<tr><td>SET_1ST_JINK</td><td>Once the evasive heading is determined and flown, the HPM starts a number of "jinks" left and right until the threat is dropped. This macro is used to determine whether the next jink is to the right or left of the evasive heading.</td><td>turn:=int_min_hdg-p_heading;
if turn>=180 then
turn:=turn-360;
if turn<(-180) then
turn:=turn+360;
if turn>=0 then
next_jink:=K_JINK_LEFT;
if turn<0 then
next_jink:=K_JINK_RIGHT;</td></tr>
<tr><td>INIT_ENTTYPE</td><td>This macro initializes constants associated with each entity type.</td><td>{init GROUND MOVER entity type constants}
K_GND_M1A1:=2001;
K_GND_T72:=3001;
K_GND_TRUCK:=3002;
K_GND_SCUD:=3003;
K_GND_T90:=3004;
K_GND_TAPZ:=3005;
K_GND_BRDM:=3006;
K_GND_BTR:=3007;
K_GND_TRACKD:=3008;
K_GND_VAN:=3009;
K_GND_MTLB:=3010;
{init BUILDING entity type constants}
K_BLD_COM:=9001;
K_BLD_MFG:=9002;
K_BLD_PWR:=9003;
K_BLD_BRIDGE:=9004;
K_BLD_SAM:=9005;
K_BLD_AIRFLD:=9006;</td></tr>
<tr><td>CALC_DETECT</td><td>This macro is used to update the moving status, gmti hit status, and size of any objects in the sensor field of view. This information, coupled</td><td>{Input: input_il_idx, il_sensor
Output: obj_detected,
set variables to examine one object from the image list at a time}</td></tr>
</table>

Macro Name	Description	Macro Code

	with the sensor used, is used to calculate the number of pixels subtended by the object on the sensor display. After calling another macro to calculate the number of pixels required to detect and identify the object, it returns whether the object is detected.	eval_type:=il_type[input_il_idx];
		{rng is converted from NM to ft for later calculation}
		eval_rng:=(il_range[input_il_idx]*6080);
		if eval_rng==0.0 then
		  eval_rng:= 999999;
		eval_moving:=il_moving[input_il_idx];
		eval_gmt_hit:=il_gmti_det[input_il_idx];
		eval_size:=il_size[input_il_idx];
		{set angles to be used if TIR is sensor used}
		tan_angle:=eval_size/eval_rng;
		obj_angle:=arctangent(tan_angle);
		{calculate pixels displayed for any given range/sensor/FOV}
		if il_sensor==K_WSAR_LOW then
		  px_displayed:=eval_size/80;
		if il_sensor==K_WSAR_MED then
		  px_displayed:=eval_size/40;
		if il_sensor==K_WSAR_LOW then
		  px_displayed:=eval_size/20;
		if il_sensor==K_NSAR_LOW then
		  px_displayed:=eval_size/10;
		if il_sensor==K_NSAR_MED then
		  px_displayed:=eval_size/3;
		if il_sensor==K_NSAR_HIGH then
		  px_displayed:=eval_size/1;
		if il_sensor==K_TIR_WIDE then
		  px_displayed:=obj_angle/0.0136;
		if il_sensor==K_TIR_NARROW then
		  px_displayed:=obj_angle/0.0068;
		if il_sensor==K_TIR_2X_NAR then
		  px_displayed:=obj_angle/0.0019;
		{for DEBUG, increase px_displayed by some factor}
		px_displayed:=px_displayed * K_PX_FACTOR;
		{call macro to determine how many pixels are required in order to detect/identify each object type}
		CALC_PIX_REQ;
		{determine whether object is detected}
		if eval_gmt_hit==TRUE | px_displayed>=px_to_detect then
		  obj_detected:=TRUE
		else
		  obj_detected:=FALSE;

CALC_IDENT	This macro is used to update the moving status, gmti hit status, and size of any objects in the sensor field	{Input: input_il_idx, il_sensor Output: obj_ident, obj_is_toi

Macro Name	Description	Macro Code

	of view. This information, coupled with the sensor used, is used to calculate the number of pixels subtended by the object on the sensor display. After calling another macro to calculate the number of pixels required to detect and identify the object, it returns whether the object is identified and whether the identified object is of the same type as the target of interest.	{set variables to examine one object from the image list at a time}
		eval_type:=il_type[input_il_idx];
		{rng is converted from NM to ft for later calculation}
		eval_rng:=(il_range[input_il_idx]*6080);
		if eval_rng==0.0 then
		  eval_rng:= 999999;
		eval_moving:=il_moving[input_il_idx];
		eval_gmt_hit:=il_gmti_det[input_il_idx];
		eval_size:=il_size[input_il_idx];
		{set angles to be used if TIR is sensor used}
		tan_angle:=eval_size/eval_rng;
		obj_angle:=arctangent(tan_angle);
		{calculate pixels displayed for any given range/sensor/FOV}
		if il_sensor==K_WSAR_LOW then
		  px_displayed:=eval_size/80;
		if il_sensor==K_WSAR_MED then
		  px_displayed:=eval_size/40;
		if il_sensor==K_WSAR_LOW then
		  px_displayed:=eval_size/20;
		if il_sensor==K_NSAR_LOW then
		  px_displayed:=eval_size/10;
		if il_sensor==K_NSAR_MED then
		  px_displayed:=eval_size/3;
		if il_sensor==K_NSAR_HIGH then
		  px_displayed:=eval_size/1;
		if il_sensor==K_TIR_WIDE then
		  px_displayed:=obj_angle/0.0136;
		if il_sensor==K_TIR_NARROW then
		  px_displayed:=obj_angle/0.0068;
		if il_sensor==K_TIR_2X_NAR then
		  px_displayed:=obj_angle/0.0019;
		{call macro to determine how many pixels are required in order to detect/identify each object type}
		CALC_PIX_REQ;
		{determine whether object is identified}
		obj_ident:=FALSE;
		obj_is_toi:=FALSE;
		if il_sensor<=K_NSAR_HIGH & eval_moving==TRUE then
		  obj_ident:=FALSE
		else
		  if px_displayed>=px_to_ident then
		    obj_ident:=TRUE;
		{determine if identified object is the target of interest}
		if obj_ident==TRUE & eval_type==K_GND_SCUD then
		  obj_is_toi:=TRUE;

CALC_NXTSNSR	This macro is used to determine the next desired sensor field of view with	{this function determines the next sensor to be used for items detected in the image.

Macro Name	Description	Macro Code

	which to view a detected object that is in TIR range.	input: il_sensor (global),
		output: nxtsnsr}
		if il_sensor<K_TIR_NARROW then
		nxtsnsr:=K_TIR_NARROW;
		if il_sensor>=K_TIR_NARROW then
		nxtsnsr:=K_TIR_2X_NAR;

CHOOSE_DEST	This macro is used to determine when the planned and updated weapon release points are sequenced. It then initiates various timers used to trigger a replan or mission abort.	{This macro is used to determine when trg_nav_nfnd and trg_nav_abrt are set.
		inputs: rp_next_step, p_ac_lat, p_ac_lon, K_ORIGWR_LAT, K_ORIGWR_LON, K_LATLON_ADJ, upd_wrp_lat, upd_wrp_lon, clock, wpt_tmpclock)
		outputs: trg_nav_nfnd, trg_nav_abrt, give_up}
		{sense when a/c is at original WRP on 1st pass}
		if rp_next_step==1 & absolute(p_ac_lat-K_ORIGWR_LAT[route]) < K_LATLON_ADJ & absolute(p_ac_lon-K_ORIGWR_LON[route]) < K_LATLON_ADJ then
		wpt_tmpclock:=clock,
		rp_next_step:=2;
		{if need a second pass, wait ~30 sec (.5 min) to make sure WRP is sequenced, then trigger nav. }
		if rp_next_step==2 & (clock-wpt_tmpclock)>=0.5 & toi_found==FALSE then
		trg_nav_nfnd:=TRUE,
		nav_nfnd_pdg:=TRUE,
		wpt_tmpclock:=10000,
		rp_next_step:=3;
		{once plan to refly is complete (where rp_next_step is set to 4 and wpt_tmpclock is reset to clock) make sure a couple of minutes have elapsed before we begin to monitor for sequencing WRP on second pass.}
		if rp_next_step==4 & (clock-wpt_tmpclock>=2.0) then
		wpt_tmpclock:=10000,
		rp_next_step:=5;
		if rp_next_step==5 & absolute(p_ac_lat-upd_wrp_lat)<K_LATLON_ADJ & absolute(p_ac_lon-upd_wrp_lon)<K_LATLON_ADJ then
		wpt_tmpclock:=clock,
		rp_next_step:=6;
		{wait 3 min to make sure WRP is sequenced & out of TIR rng, then trigger nav_abrt. in abort tasks, rp_next_step will be set to 99 such that this macro won't be called again}
		if rp_next_step==6 & (clock-wpt_tmpclock)>=3.0 then
		trg_nav_abrt:=TRUE,
		give_up:=TRUE,
		wpt_tmpclock:=10000,
		rp_next_step:=99;

CALC_PIX_REQ	The detection and identification algorithms in the model rely upon data derived from Johnson's Criteria. This approach states detection and identification criteria in terms of the number of pixels a given object subtends on a display. This macro	{input: eval_type
		output: pix_to_detect, pix_to_ident}
		px_to_detect:=1000;
		px_to_ident:=1000;
		{determine pixels needed for movers}

Macro Name	Description	Macro Code
	assigns a number of pixels required to detect and identify each of the ground objects that exists in the scenario.	if eval_type==K_GND_M1A1 then
		{calc uses T-72}
		px_to_detect:=1.7596,
		px_to_ident:=7.7064;
		if eval_type==K_GND_T72 then
		{calc uses.T-72}
		px_to_detect:=1.7596,
		px_to_ident:=7.7064;
		if eval_type==K_GND_TRUCK then
		{calc uses gen_truck}
		px_to_detect:=2.1565,
		px_to_ident:=9.3391;
		if eval_type==K_GND_SCUD then
		{calc uses TEL}
		px_to_detect:=1.4118,
		px_to_ident:=6.2756;
		if eval_type==K_GND_T90 then
		{calc uses T-90}
		px_to_detect:=1.7537,
		px_to_ident:=7.6822;
		if eval_type==K_GND_TAPZ then
		{calc uses fuel_truck}
		px_to_detect:=2.2472,
		px_to_ident:=9.7122;
		if eval_type==K_GND_BRDM then
		{calc uses BRDM}
		px_to_detect:=2.1569,
		px_to_ident:=9.3409;
		if eval_type==K_GND_BTR then
		{calc uses BTR-80}
		px_to_detect:=1.7537,
		px_to_ident:=7.6822;
		if eval_type==K_GND_TRACKD then
		{calc uses BMP}
		px_to_detect:=1.3029,
		px_to_ident:=5.8274;
		if eval_type==K_GND_VAN then
		{calc uses gen_truck}
		px_to_detect:=2.1565,
		px_to_ident:=9.3409;
		if eval_type==K_GND_MTLB then
		{calc uses MTLB_U}
		px_to_detect:=1.8250,
CALC_PIX_REQ		px_to_ident:=7.9753;
(continued)		{determine pixels needed for buildings. all calcs use cmd_post, which representys asymptote of det/ID curve}
		if eval_type==K_BLD_AIRFLD then

Macro Name	Description	Macro Code

		px_to_detect:=1.3,
		px_to_ident:=5.4;
		if eval_type==K_BLD_BRIDGE then
		px_to_detect:=1.3,
		px_to_ident:=5.4;
		if eval_type==K_BLD_COM then
		px_to_detect:=1.3,
		px_to_ident:=5.4;
		if eval_type==K_BLD_MFG then
		px_to_detect:=1.3,
		px_to_ident:=5.4;
		if eval_type==K_BLD_PWR then
		px_to_detect:=1.3,
		px_to_ident:=5.4;
		if eval_type==K_BLD_SAM then
		px_to_detect:=1.3,
		px_to_ident:=5.4;

CHK_ID_2_IMG	This macro is used to determine the next sensor look when the TIR is placed at an object's last know location, and that object is not detected. If the highest resolution sensor (TIR_2X_NAR) was used and the object is still not detected, it is assigned for shootlist removal. If an object is detected with the highest resolution sensor but can't be identified, it is classified as "unviable" and will not be looked at again until a given amount of time has passed. (Actual assignment of the unviable flag and time monitoring is performed on the FRED side.)	{Update the PL when we look at a loc at which an object is not detectable.
		NOTE: id_to_image is set to -1 for ref & for alternate pt looks.}
		found:=FALSE;
		found_status:=0;
		index:=(-1);
		if id_to_image<>(-1) then
		while index < iar_num_obj-1 do
		index+=1,
		if iar_obj_id[index] == id_to_image then
		found:=TRUE,
		found_status:=iar_obj_stat[index];
		{Update Next_Sensor in OOI Logic}
		if id_to_image<>(-1) & found==FALSE & iar_num_obj<100 & il_sensor==K_TIR_NARROW then
		{Update Next_Sensor for this object in PL}
		iar_obj_id[iar_num_obj] := id_to_image,
		iar_obj_nxts[iar_num_obj] := K_TIR_2X_NAR,
		iar_obj_stat[iar_num_obj]:=K_UPD_SENSOR,
		iar_num_obj+=1;
		{Remove from OOI Logic}
		if id_to_image<>(-1) & found==FALSE & iar_num_obj<100 & il_sensor==K_TIR_2X_NAR then
		{remove this object from PL}
		iar_obj_id[iar_num_obj] := id_to_image,
		iar_obj_stat[iar_num_obj]:=K_REMOVE_PL,
		iar_num_obj+=1;

CHK_ID_2_IMG
(continued)
		{Logic to delay looking at points that have been detected but not identified}
		index:=-1;
		l_num_iar:=iar_num_obj;

Macro Name	Description	Macro Code

		while index < l_num_iar-1 do
		index+=1,
		if iar_num_obj<100 & il_sensor==K_TIR_2X_NAR & iar_obj_stat[index]==K_DETECTED then
		{Start Unviable timer for this object in PL}
		iar_obj_id[iar_num_obj] := iar_obj_id[index],
		iar_obj_nxts[iar_num_obj] := K_TIR_NARROW,
		iar_obj_stat[iar_num_obj]:=K_UNVIABLE,
		iar_num_obj+=1;

DETECT_OBJ	This macro builds the initial list of "image analysis results". It looks at the results of the CALC_DETECT macro to determine whether each object in the image was detected.	{using Image List from FRED, this builds a list (image analysis results or 'iar') of objects detected ands marks them as 'detected'}
		{Changed 'count' to 'ct' to eliminate if character limit}
		ct:=0;
		idx:=(-1);
		while idx<il_number-1 do
		idx+=1,
		if il_prev_id[idx]==FALSE&(il_sensor<>K_REAL_BEAM | (il_sensor==K_REAL_BEAM & il_rng_ref[idx]<K_RBEAM_ROI)) then
		input_il_idx:=idx,
		CALC_DETECT,
		if obj_detected==TRUE then
		iar_obj_mvng[ct]:=il_moving[idx],
		cns_moving:=iar_obj_mvng[ct],
		CALC_NXTSNSR,
		iar_obj_id[ct]:=il_obj_id[idx],
		iar_obj_lat[ct]:=il_lat[idx],
		iar_obj_lon[ct]:=il_lon[idx],
		iar_obj_stat[ct]:=K_DETECTED,
		iar_obj_nxts[ct]:=nxtsnsr,
		detect_il[ct]:=idx,
		ct+=1;
		{store number of entries placed in the table}
		iar_num_obj:=ct;
		{store the number that were detected--this will be adjusted for any objects that are subsequently identified}
		iar_num_det:=ct;

CLR_ACQ_TRG	This macro is used at the end of the Acquisition function to reset the Acquisition triggers.	trg_acq_rb:=FALSE; trg_acq_upd:=FALSE;
CLR_ATK_TRG	This macro is used at the end of the Attack function to reset the Attack trigger.	trg_attack:=FALSE;
CLR_EVD_TRG	This macro is used at the end of the Evade function to reset the Evade trigger.	trg_evade:=FALSE;

CLR_NAV_TRG	This macro is called at the end of the Navigate goal function to reset the Navigate triggers.	trg_nav_abrt:=FALSE;
		trg_nav_fnd:=FALSE;
		trg_nav_pln:=FALSE;
		trg_nav_nfnd:=FALSE;

Macro Name	Description	Macro Code

RANK_THREATS	This macro is used to rank threats launched simultaneously against the aircraft. It bases the ranking on time to intercept (TTI) which is passed to the HPM by FRED. While TTI would not be explicitly available to the pilot, it is assumed that it could be estimated based on the time and range of the launches (info that is available).	{If there are multiple launches that can kill you, rank them based on time to intercept to find the most dangerous (top_threat)}
		tht_index:=0;
		shortest_tti:=1000.0;
		while tht_index<tht_count do
		  if bad_tht[tht_index]==TRUE & tht_msltti[tht_index]<shortest_tti then
		    shortest_tti:=tht_msltti[tht_index],
		    top_threat:=tht_index,
		    curr_emitter:=tht_emtid[tht_index],
		  tht_index+=1
		else
		  tht_index+=1;

IDENTIFY_OBJ	This macro builds the final list of "image analysis results". It looks at the results of the CALC_IDENT macro to determine whether each object in the image was identified.	{looks at image list from FRED, iar data from Detect_OBJ, detect_il, and builds a list of objects (iar) that were ID'd. It also identifies if the TOI has been found}
		count:=0;
		toi_found:=FALSE;
		idx:=-1;
		while idx<iar_num_obj-1 do
		  idx+=1,
		  input_il_idx:=detect_il[idx],
		  CALC_IDENT,
		    if obj_ident==TRUE then
		    iar_obj_stat[idx]:=K_IDENTIFIED,
		    count+=1,
		    if obj_is_toi==TRUE then
		      toi_found:=TRUE,
		      toi_index:=detect_il[idx];
		{code does not yet allow mis-identification of objects}
		{store the number of objects ID'd}
		iar_num_id:=count;
		{reset the number detected based on number id'd}
		iar_num_det:=iar_num_det-count;
		{set sensor used}
		iar_sensor:=il_sensor;

REF_IN_RNG	This macro looks at the range and bearing to the reference to determine whether it is within the viewable region of the SAR and/or TIR.	{determine if current ref point is in rng to image with TIR and/or SAR.
		Inputs: p_rng_to_ref, p_brg_to_ref.
		Outputs: in_sar_rng, in_tir_rng, ref_in_rng}
		if p_rng_to_ref<=K_SAR_RNG & absolute(p_brg_to_ref) >=K_SAR_GIM_LO & absolute(p_brg_to_ref) <= K_SAR_GIM_HI then
		  in_sar_rng:=TRUE
		else
		  in_sar_rng:=FALSE;

Macro Name	Description	Macro Code

		if p_rng_to_ref<=K_TIR_RNG then
		in_tir_rng:=TRUE
		else
		in_tir_rng:=FALSE;
		if in_tir_rng==TRUE | in_sar_rng==TRUE then
		ref_in_rng:=TRUE
		else
		ref_in_rng:=FALSE;

INIT_ALT_ACQ	This macro is used to initialize the alternate waypoints for each mission. If the original weapon release point is sequenced and the target is not yet found, these alternate waypoints will be entered into the planner prior to requesting a replan to refly the target area.	K_ALT_PT_LAT[1,1]:=46.3; K_ALT_PT_LON[1,1]:=23.0; K_ALT_PT_LAT[1,2]:=46.2; K_ALT_PT_LON[1,2]:=22.7; K_ALT_PT_LAT[2,1]:=45.7; K_ALT_PT_LON[2,1]:=23.4; K_ALT_PT_LAT[2,2]:=45.9; K_ALT_PT_LON[2,2]:=23.1; K_ALT_PT_LAT[3,1]:=45.6; K_ALT_PT_LON[3,1]:=22.3; K_ALT_PT_LAT[3,2]:=45.5; K_ALT_PT_LON[3,2]:=22.7; K_ALT_PT_LAT[4,1]:=46.4; K_ALT_PT_LON[4,1]:=22.8; K_ALT_PT_LAT[4,2]:=46.2; K_ALT_PT_LON[4,2]:=22.6; K_ALT_PT_LAT[5,1]:=45.57; K_ALT_PT_LON[5,1]:=22.07; K_ALT_PT_LAT[5,2]:=45.71; K_ALT_PT_LON[5,2]:=23.00; K_ALT_PT_LAT[6,1]:=46.4; K_ALT_PT_LON[6,1]:=22.45; K_ALT_PT_LAT[6,2]:=46.35; K_ALT_PT_LON[6,2]:=22.8; {initialize the number of alt waypoints for each of the six routes} K_NUM_ALT_PT[1]:=2; K_NUM_ALT_PT[2]:=2; K_NUM_ALT_PT[3]:=2; K_NUM_ALT_PT[4]:=2; K_NUM_ALT_PT[5]:=2; K_NUM_ALT_PT[6]:=2;
INIT_CMD_IDS	This macro specifies the enumeration for each of the HPM commands given to FRED.	K_REL_WCM:=1; K_GAMEOVER:=15; K_UFC_PLNMOD:=2; K_UFC_TGTMOD:=3; K_DOI_LFTMPD:=11; K_DOI_RTMPD:=12; K_DOI_CTRMPD:=10; K_CURSOR_AC:=8;

Macro Name	Description	Macro Code
		K_DESIG_PT:=13; K_MAK_MSTFLY:=14; K_MOVECURSOR:=9; K_F_ACP_RPLN:=6; K_REQ_RPLN:=4; K_U_ACP_RPLN:=7; K_REQ_ABORT:=5; K_DEPLOY_TIR:=24; K_STOW_TIR:=23; K_A_PILOT_D:=21; K_A_PILOT_E:=20; K_RESET_SL:=22; K_REL_CMEAS:=16; K_EVADE_MNVR:=25; K_THROTTLE:=19; K_A_THROTTLE:=18; K_RMV_SL_OBJ:=17; K_UFC_TGT_LL:=27; K_UFC_LL_ACP:=26; K_UPD_REFLOC:=29; K_RECALC_PL:=33; K_RESET_OOI:=32; K_CALC_ALTPT:=34; K_GEN_IMAGE:=28; K_PROCES_OOI:=30; K_RANK_OOI:=31;
INIT_GLOBAL INIT_GLOBAL (continued)	This macro initializes all of the scenario-independent variables prior to the start of a trial.	{set other constants} K_DETECTED:=1; K_IDENTIFIED:=2; K_REMOVE_PL:=3; K_UPD_SENSOR:=4; K_UNVIABLE:=5; K_EMT_TYPE_1:=7405; K_EMT_TYPE_2:=7015; K_TIR_2X_NAR:=9; K_TIR_NARROW:=8; K_TIR_WIDE:=7; K_NSAR_HIGH:=6; K_NSAR_MED:=5; K_NSAR_LOW:=4; K_WSAR_HIGH:=3; K_WSAR_MED:=2; K_WSAR_LOW:=1; K_REAL_BEAM:=0; K_DELAY_END:=1; K_LAR_RNG:=6; K_HOME_LAT:=43.50;

Macro Name	Description	Macro Code

INIT_GLOBAL (continued)		K_HOME_LON:=16.29999917;
		K_SAR_RNG:=70;
		K_TIR_RNG:=20;
		K_SAR_GIM_HI:=60;
		K_SAR_GIM_LO:=0;
		K_DFLT_SNSR:=K_WSAR_LOW;
		K_BINGO_FUEL:=2000;
		K_RP_TYP_REQ:=0;
		K_RP_TYP_NAV:=2;
		K_RP_TYP_THT:=3;
		K_RP_TYP_FUL:=4;
		K_RP_TYP_ERR:=5;
		K_LATLON_ADJ:=.04;
		K_DOI_LEFT:=1;
		K_DOI_CENTER:=2;
		K_DOI_RIGHT:=3;
		K_UFC_PLAN:=1;
		K_UFC_NAV:=2;
		K_UFC_TGT:=3;
		K_HDG_ADJ:=5.0;
		K_JINK_LEFT:=1;
		K_JINK_RIGHT:=2;
		K_MAX_ALT_LK:=5;
		K_PX_FACTOR:=1;
		K_RBEAM_ROI:=10;
		TRUE:=1;
		FALSE:=0;
		ON:=1;
		OFF:=0;
		SAR:=0;
		TIR:=0;
		CRUISE:=1;
		FULL_MIL:=2;
		AFTERBURNER:=3;
		AUTOTHROTTLE:=4;
		{initialize variables for start of mission}
		hpm_cmd_id:=K_PROCES_OOI;
		hpm_rti_trig+=1;
		current_doi:=2;
		auto_pilot:=ON;
		must_evade:=FALSE;
		cur_throttle:=CRUISE;
		tgt_update:=FALSE;
		pl_number:=0;
		index_to_alt:=0;
		pass:=1;
		start_timer:=10000;

Macro Name	Description	Macro Code

		next_jink:=0;
		need_reflook:=TRUE;
		lastpl_count:=(-1);
		ufc_mode:=K_UFC_PLAN;
		pre_tir_upd:=FALSE;
		pre_tir_pndg:=FALSE;
		rp_next_step:=1;
		alt_pt_index:=1;
		wpt_tmpclock:=10000;

INIT_LK_PT1	This macro initializes the alternate lookpoints around the original and updated reference points for Scenario 1. These points were created a prior using a map of the gaming area. They represent points that lay on road networks surrounding both the planned and updated reference points.	{Define the look points around the initial and updated target coordinates for Route 1. Uses 3D array, [route number, initial (1) v. updated (2), look number (0-4)} K_LK_PT_LAT[1,1,0]:= 45.86670; K_LK_PT_LAT[1,1,1]:= 45.84833; K_LK_PT_LAT[1,1,2]:= 45.85217; K_LK_PT_LAT[1,1,3]:= 45.83000; K_LK_PT_LAT[1,1,4]:= 45.82583; K_LK_PT_LON[1,1,0]:= 22.90000; K_LK_PT_LON[1,1,1]:= 22.90733; K_LK_PT_LON[1,1,2]:= 22.904083; K_LK_PT_LON[1,1,3]:= 22.94633; K_LK_PT_LON[1,1,4]:= 22.91800; K_LK_PT_LAT[1,2,0]:= 45.85770; K_LK_PT_LAT[1,2,1]:= 45.85833; K_LK_PT_LAT[1,2,2]:= 45.84167; K_LK_PT_LAT[1,2,3]:= 45.83083; K_LK_PT_LAT[1,2,4]:= 45.83500; K_LK_PT_LON[1,2,0]:= 22.91490; K_LK_PT_LON[1,2,1]:= 22.94333; K_LK_PT_LON[1,2,2]:= 22.89500; K_LK_PT_LON[1,2,3]:= 22.92083; K_LK_PT_LON[1,2,4]:= 22.95167;
INIT_LK_PT2	This macro initializes the alternate lookpoints around the original and updated reference points for Scenario 2. These points were created a prior using a map of the gaming area. They represent points that lay on road networks surrounding both the planned and updated reference points.	{Define the alternate look points around the initial and updated target coordinates for Route 2. Uses 3D array, [route number, initial (1) v. updated (2), look_pt (0-4)} K_LK_PT_LAT[2,1,0]:= 45.90000; K_LK_PT_LAT[2,1,1]:= 45.91550; K_LK_PT_LAT[2,1,2]:= 45.90500; K_LK_PT_LAT[2,1,3]:= 45.91500; K_LK_PT_LAT[2,1,4]:= 45.91167; K_LK_PT_LON[2,1,0]:= 22.75000; K_LK_PT_LON[2,1,1]:= 22.73333; K_LK_PT_LON[2,1,2]:= 22.73700; K_LK_PT_LON[2,1,3]:= 22.76167; K_LK_PT_LON[2,1.4]:= 22.78167; K_LK_PT_LAT[2,2.0]:= 45.89180;

Macro Name	Description	Macro Code
		K_LK_PT_LAT[2,2,1]:= 45.90000; K_LK_PT_LAT[2,2,2]:= 45.91000; K_LK_PT_LAT[2,2,3]:= 45.91500; K_LK_PT_LAT[2,2,4]:= 45.91167; K_LK_PT_LON[2,2,0]:= 22.74610; K_LK_PT_LON[2,2,1]:= 22.73800; K_LK_PT_LON[2,2,2]:= 22.73800; K_LK_PT_LON[2,2,3]:= 22.76167; K_LK_PT_LON[2,2,4]:= 22.78167;
INIT_LK_PT3	This macro initializes the alternate lookpoints around the original and updated reference points for Scenario 3. These points were created a prior using a map of the gaming area. They represent points that lay on road networks surrounding both the planned and updated reference points.	{Define the alternate look points around the initial and updated target coordinates for Route 3. Uses 3D array, [route number, initial (1) v. updated (2), look_pt (0-4)} K_LK_PT_LAT[3,1,0]:= 45.88333; K_LK_PT_LAT[3,1,1]:= 45.89500; K_LK_PT_LAT[3,1,2]:= 45.90417; K_LK_PT_LAT[3,1,3]:= 45.90667; K_LK_PT_LAT[3,1,4]:= 45.91000; K_LK_PT_LON[3,1,0]:= 22.88333; K_LK_PT_LON[3,1,1]:= 22.89833; K_LK_PT_LON[3,1,2]:= 22.88167; K_LK_PT_LON[3,1,3]:= 22.87500; K_LK_PT_LON[3,1,4]:= 22.87500; K_LK_PT_LAT[3,2,0]:= 45.89520; K_LK_PT_LAT[3,2,1]:= 45.88333; K_LK_PT_LAT[3,2,2]:= 45.90833; K_LK_PT_LAT[3,2,3]:= 45.91000; K_LK_PT_LAT[3,2,4]:= 45.89333; K_LK_PT_LON[3,2,0]:= 22.88200; K_LK_PT_LON[3,2,1]:= 22.87667; K_LK_PT_LON[3,2,2]:= 22.87333; K_LK_PT_LON[3,2,3]:= 22.90333; K_LK_PT_LON[3,2,4]:= 22.90500;
INIT_LK_PT4	This macro initializes the alternate lookpoints around the original and updated reference points for Scenario 4. These points were created a prior using a map of the gaming area. They represent points that lay on road networks surrounding both the planned and updated reference points.	{Define the alternate look points around the initial and updated target coordinates for Route 4. Uses 3D array, [route number, initial (1) v. updated (2), look_pt (0-4)} K_LK_PT_LAT[4,1,0]:= 45.91660; K_LK_PT_LAT[4,1,1]:= 45.92217; K_LK_PT_LAT[4,1,2]:= 45.93333; K_LK_PT_LAT[4,1,3]:= 45.94850; K_LK_PT_LAT[4,1,4]:= 45.94500; K_LK_PT_LON[4,1,0]:= 22.76660; K_LK_PT_LON[4,1,1]:= 22.76867; K_LK_PT_LON[4,1,2]:= 22.75167; K_LK_PT_LON[4,1,3]:= 22.76417; K_LK_PT_LON[4,1,4]:= 22.78667; K_LK_PT_LAT[4,2,0]:= 45.94490; K_LK_PT_LAT[4,2,1]:= 45.94800;

Macro Name	Description	Macro Code
		K_LK_PT_LAT[4,2,2]:= 45.95167; K_LK_PT_LAT[4,2,3]:= 45.93500; K_LK_PT_LAT[4,2,4]:= 45.93333; K_LK_PT_LON[4,2,0]:= 22.79290; K_LK_PT_LON[4,2,1]:= 22.77500; K_LK_PT_LON[4,2,2]:= 22.76000; K_LK_PT_LON[4,2,3]:= 22.75000; K_LK_PT_LON[4,2,4]:= 22.78167;
INIT_LK_PT5	This macro initializes the alternate lookpoints around the original and updated reference points for Scenario 5. These points were created a prior using a map of the gaming area. They represent points that lay on road networks surrounding both the planned and updated reference points.	{Define the alternate look points around the initial and updated target coordinates for Route 5. Uses 3D array, [route number, initial (1) v. updated (2), look_pt (0-4)} K_LK_PT_LAT[5,1,0]:= 45.88330; K_LK_PT_LAT[5,1,1]:= 45.90000; K_LK_PT_LAT[5,1,2]:= 45.86583; K_LK_PT_LAT[5,1,3]:= 45.84433; K_LK_PT_LAT[5,1,4]:= 45.84167; K_LK_PT_LON[5,1,0]:= 22.80000; K_LK_PT_LON[5,1,1]:= 22.80500; K_LK_PT_LON[5,1,2]:= 22.79833; K_LK_PT_LON[5,1,3]:= 22.81500; K_LK_PT_LON[5,1,4]:= 22.83500; K_LK_PT_LAT[5,2,0]:= 45.85160; K_LK_PT_LAT[5,2,1]:= 45.85833; K_LK_PT_LAT[5,2,2]:= 45.85333; K_LK_PT_LAT[5,2,3]:= 45.84333; K_LK_PT_LAT[5,2,4]:= 45.84750; K_LK_PT_LON[5,2,0]:= 22.79110; K_LK_PT_LON[5,2,1]:= 22.79333; K_LK_PT_LON[5,2,2]:= 22.80000; K_LK_PT_LON[5,2,3]:= 22.82667; K_LK_PT_LON[5,2,4]:= 22.80333;
INIT_LK_PT6	This macro initializes the alternate lookpoints around the original and updated reference points for Scenario 6. These points were created a prior using a map of the gaming area. They represent points that lay on road networks surrounding both the planned and updated reference points.	{Define the alternate look points around the initial and updated target coordinates for Route 6. Uses 3D array, [route number, initial (1) v. updated (2), look_pt (0-4)} K_LK_PT_LAT[6,1,0]:= 45.95000; K_LK_PT_LAT[6,1,1]:= 45.93500; K_LK_PT_LAT[6,1,2]:= 45.94500; K_LK_PT_LAT[6,1,3]:= 45.96450; K_LK_PT_LAT[6,1,4]:= 45.97833; K_LK_PT_LON[6,1,0]:= 22.76670; K_LK_PT_LON[6,1,1]:= 22.75500; K_LK_PT_LON[6,1,2]:= 22.78917; K_LK_PT_LON[6,1,3]:= 22.78800; K_LK_PT_LON[6,1,4]:= 22.80333; K_LK_PT_LAT[6,2,0]:= 45.95780; K_LK_PT_LAT[6,2,1]:= 45.97000; K_LK_PT_LAT[6,2,2]:= 45.97833;

Macro Name	Description	Macro Code
		K_LK_PT_LAT[6,2,3]:= 45.98167; K_LK_PT_LAT[6,2,4]:= 45.97000; K_LK_PT_LON[6,2,0]:= 22.78150; K_LK_PT_LON[6,2,1]:= 22.79667; K_LK_PT_LON[6,2,2]:= 22.80333; K_LK_PT_LON[6,2,3]:= 22.78833; K_LK_PT_LON[6,2,4]:= 22.76833;

INIT_SAR_LKQ
 	
INIT_SAR_LKQ
(continued)
	The model was built on the assumption that the SAR leg of the acquisition phase would be used only for object detection and building up a shootlist. This macro defines the sequecence of SAR images to be taken during that leg. It consists of 16 images that include the reference point and four alternate lookpoints at multiple resolutions.	{Initialize the SAR Look Points Queue}
		{SAR Look Points - Use same Lat/Long index for Initial & Update}
		K_S_LP_MAX:=16;
		{For *IDX, 0=>Ref Pt Coord, 1-4=>Alt Pt 1-4 Coord}
		K_S_LP_IDX[0]:=0;
		K_S_LP_IDX[1]:=0;
		K_S_LP_IDX[2]:=0;
		K_S_LP_IDX[3]:=1;
		K_S_LP_IDX[4]:=1;
		K_S_LP_IDX[5]:=2;
		K_S_LP_IDX[6]:=2;
		K_S_LP_IDX[7]:=3;
		K_S_LP_IDX[8]:=3;
		K_S_LP_IDX[9]:=4;
		K_S_LP_IDX[10]:=4;
		K_S_LP_IDX[11]:=0;
		K_S_LP_IDX[12]:=1;
		K_S_LP_IDX[13]:=2;
		K_S_LP_IDX[14]:=3;
		K_S_LP_IDX[15]:=4;
		K_S_LP_SNSR[0]:=K_REAL_BEAM;
		K_S_LP_SNSR[1]:=K_WSAR_MED;
		K_S_LP_SNSR[2]:=K_NSAR_LOW;
		K_S_LP_SNSR[3]:=K_WSAR_MED;
		K_S_LP_SNSR[4]:=K_NSAR_LOW;
		K_S_LP_SNSR[5]:=K_WSAR_MED;
		K_S_LP_SNSR[6]:=K_NSAR_LOW;
		K_S_LP_SNSR[7]:=K_WSAR_MED;
		K_S_LP_SNSR[8]:=K_NSAR_LOW;
		K_S_LP_SNSR[9]:=K_WSAR_MED;
		K_S_LP_SNSR[10]:=K_NSAR_LOW;
		K_S_LP_SNSR[11]:=K_REAL_BEAM;
		K_S_LP_SNSR[12]:=K_WSAR_MED;
		K_S_LP_SNSR[13]:=K_WSAR_MED;
		K_S_LP_SNSR[14]:=K_WSAR_MED;
		K_S_LP_SNSR[15]:=K_WSAR_MED;
		next_isar_lp:=0;
		next_usar_lp:=0;

INIT_SPECIFC	This macro initializes the lat/lon of the original weapon release point and the "known" lat/lon of the target of	{define lat lon of planned weapon release point} K_ORIGWR_LAT[1]:=45.86666472;

Macro Name	Description	Macro Code

INIT_SPECIFC (continued)	interest at the time of mission planning. This lat/lon value is used to set the initial hpm_ref_lat and hpm_ref_lon variables. The lat/longs include scenario specific values for each of six scenarios.	K_ORIGWR_LON[1]:=22.89999944;
		{K_ORIGWR_LAT[1]:=45.689;
		K_ORIGWR_LON[1]:=22.533;}
		K_ORIGWR_LAT[2]:=45.90000139;
		K_ORIGWR_LON[2]:=22.75;
		K_ORIGWR_LAT[3]:=45.883335;
		K_ORIGWR_LON[3]:=22.88333306;
		K_ORIGWR_LAT[4]:=45.9166;
		K_ORIGWR_LON[4]:=22.7666;
		{[5] was 45.87284833, 22.6121805}
		K_ORIGWR_LAT[5]:=45.8544;
		K_ORIGWR_LON[5]:=22.7241;
		K_ORIGWR_LAT[6]:=45.91619861;
		K_ORIGWR_LON[6]:=22.82497;
		{define lat lon of waypoint that follows the planned weapon release point – COMMENTED OUT!}
		{K_POSTWR_LAT[1]:=43.5;
		K_POSTWR_LON[1]:=16.29999917;
		K_POSTWR_LAT[2]:=45.95697;
		K_POSTWR_LON[2]:=21.20853972;
		K_POSTWR_LAT[3]:=43.5;
		K_POSTWR_LON[3]:=16.29999917;
		K_POSTWR_LAT[4]:=45.62026972;
		K_POSTWR_LON[4]:=21.91443806;
		K_POSTWR_LAT[5]:=45.77896861;
		K_POSTWR_LON[5]:=22.59836944;
		K_POSTWR_LAT[6]:=43.5;
		K_POSTWR_LON[6]:=16.29999917;}
		{define lat lon of target as known at the time of mission planning}
		K_TGTBEG_LAT[1]:=45.8667;
		K_TGTBEG_LON[1]:=22.9;
		K_TGTBEG_LAT[2]:=45.9;
		K_TGTBEG_LON[2]:=22.75;
		K_TGTBEG_LAT[3]:=45.8833;
		K_TGTBEG_LON[3]:=22.8833;
		K_TGTBEG_LAT[4]:=45.9166;
		K_TGTBEG_LON[4]:=22.7666;
		K_TGTBEG_LAT[5]:=45.8833;
		K_TGTBEG_LON[5]:=22.8;
		K_TGTBEG_LAT[6]:=45.9333;
		K_TGTBEG_LON[6]:=22.75;
		{set initial reference point based on route (from FRED)}
		hpm_ref_lat:=K_TGTBEG_LAT[route];
		hpm_ref_lon:=K_TGTBEG_LON[route];

INIT_TIR_LKQ	Sensor employment during the TIR leg of the acquisition cycle is intended primarily for target identification. The model will attempt to use the TIR to examine and	{Initialize the TIR Look Points Queue} {TIR Look Points - Use same Lat/Long index for Initial & Update} K_T_LP_MAX:=10;

Macro Name	Description	Macro Code

	identlfy anything that was detected and added to the shootlist during the SAR leg. If however, there are no viable objects on the shootlist at the time the aircraft is in TIR range to the reference point, the TIR will be used to examine the reference point and the four alternate look points. This macro specifies the sequence of these TIR looks.	{For *IDX, 0=>Ref Pt Coord, 1-4=>Alt Pt 1-4 Coord}
		K_T_LP_IDX[0]:=0;
		K_T_LP_IDX[1]:=0;
		K_T_LP_IDX[2]:=1;
		K_T_LP_IDX[3]:=1;
		K_T_LP_IDX[4]:=2;
		K_T_LP_IDX[5]:=2;
		K_T_LP_IDX[6]:=3;
		K_T_LP_IDX[7]:=3;
		K_T_LP_IDX[8]:=4;
		K_T_LP_IDX[9]:=4;
		K_T_LP_SNSR[0]:=K_TIR_NARROW;
		K_T_LP_SNSR[1]:=K_TIR_2X_NAR;
		K_T_LP_SNSR[2]:=K_TIR_NARROW;
		K_T_LP_SNSR[3]:=K_TIR_2X_NAR;
		K_T_LP_SNSR[4]:=K_TIR_NARROW;
		K_T_LP_SNSR[5]:=K_TIR_2X_NAR;
		K_T_LP_SNSR[6]:=K_TIR_NARROW;
		K_T_LP_SNSR[7]:=K_TIR_2X_NAR;
		K_T_LP_SNSR[8]:=K_TIR_NARROW;
		K_T_LP_SNSR[9]:=K_TIR_2X_NAR;
		next_itir_lp:=0;
		next_utir_lp:=0;

SET_SAR_LKPT	This macro is used to set a SAR look pointer to either the original or updated reference and alternate lookpoints, based on whether the target update has been completed. In addition it sets a flag if the current look point is an alternate lookpoint. (This has implications for display of interest selection when imaging).	if tgt_upd_done==TRUE then
		{Use Update Queue}
		idx:=K_S_LP_IDX[next_usar_lp],
		{idx now contains pointer to lat/long of ref/alt look pts}
		{Note: The '2' indicates use of the update coord}
		lat_to_image:=K_LK_PT_LAT[route,2,idx],
		lon_to_image:=K_LK_PT_LON[route,2,idx],
		sensor_2_use:= K_S_LP_SNSR[next_usar_lp],
		next_usar_lp+=1
		else
		{Use Initial Queue}
		idx:=K_S_LP_IDX[next_isar_lp],
		{idx now contains pointer to lat/long of ref/alt look pts}
		{Note: The '1' indicates use of the initial coord}
		lat_to_image:=K_LK_PT_LAT[route,1,idx],
		lon_to_image:=K_LK_PT_LON[route,1,idx],
		sensor_2_use:= K_S_LP_SNSR[next_isar_lp],
		next_isar_lp+=1;
		{Set use of Alt Pt or not}
		if idx==0 then
		alt_pt_inuse:=FALSE
		else
		alt_pt_inuse:=TRUE;

Macro Name	Description	Macro Code

		{Handle wraparound}
		if next_usar_lp >= K_S_LP_MAX then
		  next_usar_lp:=11;
		if next_isar_lp >= K_S_LP_MAX then
		  next_isar_lp:=11;

SET_TIR_LKPT	This macro is used to set a TIR look pointer to either the original or updated reference and alternate lookpoints, based on whether the target update has been completed. In addition it sets a flag if the current look point is an alternate lookpoint. (This has implications for display of interest selection when imaging).	if tgt_upd_done==TRUE then
		  {Use Update Queue}
		  idx:=K_T_LP_IDX[next_utir_lp],
		  {idx now contains pointer to lat/long of ref/alt look pts}
		  {Note: The '2' indicates use of the update coord}
		  lat_to_image:=K_LK_PT_LAT[route,2,idx],
		  lon_to_image:=K_LK_PT_LON[route,2,idx] ,
		  sensor_2_use:= K_T_LP_SNSR[next_utir_lp],
		  next_utir_lp+=1
		else
		  {Use Initial Queue}
		  idx:=K_T_LP_IDX[next_itir_lp],
		  {idx now contains pointer to lat/long of ref/alt look pts}
		  {Note: The '1' indicates use of the initial coord}
		  lat_to_image:=K_LK_PT_LAT[route,1,idx],
		  lon_to_image:=K_LK_PT_LON[route,1,idx],
		  sensor_2_use:= K_T_LP_SNSR[next_itir_lp],
		  next_itir_lp+=1;

SET_TIR_LKPT (continued)		{Set use of Alt Pt or not}
		if idx==0 then
		  alt_pt_inuse:=FALSE
		else
		  alt_pt_inuse:=TRUE;
		{Handle wraparound}
		if next_utir_lp >= K_T_LP_MAX then
		  next_utir_lp:=0;
		if next_itir_lp >= K_T_LP_MAX then
		  next_itir_lp:=0;

APPENDIX H

SAMPLE CALCULATION OF MEAN RANKS

This page left blank intentionally

Table H-1. Sample Calculation of Mean Ranks

Group	Original Results	Tie Together Identical Results & Sort in Ascending Order	Rank position	Mean Rank (Sum the Rank Positions and Divide by Number of Ties)	Assign Mean Ranks to Original Results
A	15	[redacted]	[redacted]	[redacted]	6
A	10	[redacted]	[redacted]	[redacted]	4
A	[redacted]	10	3	4	[redacted]
A	20	10	4	4	7.5
B	20	10	5	4	7.5
B	[redacted]	15	6	6	[redacted]
B	10	20	7	7.5	4
B	10	20	8	7.5	4

Ranks 1+2=3
Divide by 2 ties=1.5

Group A Average Rank $= \dfrac{19}{4} = 4.75$

Group B Average Rank $= \dfrac{17}{4} = 4.25$

This page left blank intentionally

APPENDIX I

AVERAGE MEAN RANKS

This page left blank intentionally

Table I-1. Average Mean Ranks

Scenario #				# of nav error-related replans generated	number of nav error-related replans accepted	number of threat-based replans generated	number of threat-based replans accepted	number of missiles locked on	number of missiles launched	percent of missiles defeated	range at release
1	Operator	HITL	Mean	46.2500	44.6875	39.0000	41.0000	38.0625	31.8125	50.1875	47.7500
			Std Deviation	23.8253	24.4130	.0000	.0000	17.1453	18.3321	21.7878	25.5609
			Minimum	14.0000	17.5000	39.0000	41.0000	18.0000	17.5000	10.5000	1.0000
			Maximum	73.0000	75.0000	39.0000	41.0000	62.5000	66.5000	77.0000	76.0000
		HPM	Mean	33.3333	37.8333	69.8333	71.0000	43.0833	34.1667	34.3333	40.6667
			Std Deviation	14.9755	15.7501	15.1052	14.6969	6.5147	8.0042	12.2787	13.2765
			Minimum	14.0000	17.5000	39.0000	41.0000	32.0000	29.0000	10.5000	22.0000
			Maximum	43.0000	48.0000	76.0000	77.0000	48.0000	44.5000	41.0000	61.0000
2	Operator	HITL	Mean	67.0625	59.9375	34.8125	36.6875	33.8125	48.0000	22.4375	32.0000
			Std Deviation	12.1080	20.9087	11.8440	12.1976	17.9601	6.4807	20.9292	25.7793
			Minimum	43.0000	17.5000	5.5000	6.5000	13.5000	44.5000	10.5000	4.0000
			Maximum	81.0000	81.0000	39.0000	41.0000	65.5000	58.5000	70.0000	70.0000
		HPM	Mean	73.8333	75.2500	39.0000	41.0000	67.9167	67.5833	31.8333	24.1667
			Std Deviation	8.6062	7.2509	.0000	.0000	17.0570	7.3104	19.6638	16.2655
			Minimum	61.0000	64.0000	39.0000	41.0000	39.0000	58.5000	10.5000	2.0000
			Maximum	83.5000	83.5000	39.0000	41.0000	84.0000	81.0000	56.5000	52.0000

Table I-1. Average Mean Ranks (continued)

Scenario #				# of nav error-related replans generated	number of nav error-related replans accepted	number of threat-based replans generated	number of threat-based replans accepted	number of missiles locked on	number of missiles launched	percent of missiles defeated	range at release
3	Operator	HITL	Mean	51.0000	54.4375	35.2500	36.8750	58.1250	60.1875	41.2500	51.3750
			Std Deviation	21.6647	20.2933	22.3639	22.4893	28.1599	33.0410	21.0527	30.7475
			Minimum	14.0000	17.5000	5.5000	6.5000	13.5000	7.5000	10.5000	3.0000
			Maximum	77.5000	78.0000	76.0000	77.0000	81.0000	84.0000	66.5000	83.0000
		HPM	Mean	14.0000	17.5000	5.5000	6.5000	7.2500	5.6667	77.0000	53.1667
			Std Deviation	.0000	.0000	.0000	.0000	3.0619	2.8402	.0000	10.9621
			Minimum	14.0000	17.5000	5.5000	6.5000	6.0000	2.0000	77.0000	39.0000
			Maximum	14.0000	17.5000	5.5000	6.5000	13.5000	7.5000	77.0000	65.0000
4	Operator	HITL	Mean	28.5000	21.3125	48.2500	36.6875	9.0000	23.1875	58.6429	53.0000
			Std Deviation	15.5012	10.7834	17.1277	12.1976	6.1644	18.3944	22.8194	22.6968
			Minimum	14.0000	17.5000	39.0000	6.5000	1.0000	2.0000	10.5000	9.0000
			Maximum	43.0000	48.0000	76.0000	41.0000	18.0000	44.5000	77.0000	80.0000
		HPM	Mean	41.1667	45.5833	57.5000	59.0000	35.0833	23.9167	25.8333	43.1667
			Std Deviation	15.1316	15.1737	20.2657	19.7180	11.6207	11.0834	23.7543	19.7729
			Minimum	14.0000	17.5000	39.0000	41.0000	24.5000	17.5000	10.5000	8.0000
			Maximum	61.0000	64.0000	76.0000	77.0000	58.0000	44.5000	56.5000	63.0000

Table I-1. Average Mean Ranks (continued)

Scenario #				# of nav error-related replans generated	number of nav error-related replans accepted	number of threat-based replans generated	number of threat-based replans accepted	number of missiles locked on	number of missiles launched	percent of missiles defeated	range at release
5	Operator	HITL	Mean	47.6250	45.1250	39.0000	41.0000	62.1250	61.9375	35.6250	47.2500
			Std Deviation	17.5087	19.0277	.0000	.0000	17.2249	17.9552	18.9732	27.7682
			Minimum	14.0000	17.5000	39.0000	41.0000	39.0000	29.0000	10.5000	13.0000
			Maximum	67.0000	70.0000	39.0000	41.0000	82.0000	78.5000	62.5000	82.0000
		HPM	Mean	46.4167	49.2500	39.0000	41.0000	54.6667	54.5833	62.7500	29.5000
			Std Deviation	30.6470	28.9840	.0000	.0000	20.7814	18.7947	16.4370	11.2739
			Minimum	14.0000	17.5000	39.0000	41.0000	24.5000	29.0000	36.0000	16.0000
			Maximum	83.5000	83.5000	39.0000	41.0000	76.5000	78.5000	77.0000	50.0000
6	Operator	HITL	Mean	38.7500	38.5625	39.4375	36.8750	55.5000	54.1250	33.5000	48.0000
			Std Deviation	17.3761	18.2609	18.8593	22.4893	18.4372	23.3663	21.2620	28.2725
			Minimum	14.0000	17.5000	5.5000	6.5000	18.0000	17.5000	10.5000	14.0000
			Maximum	67.0000	64.0000	76.0000	77.0000	79.0000	75.5000	70.0000	79.0000
		HPM	Mean	14.0000	17.5000	69.8333	71.0000	44.8333	36.7500	35.0000	25.8333
			Std Deviation	.0000	.0000	15.1052	14.6969	16.1823	8.4897	17.4011	29.0339
			Minimum	14.0000	17.5000	39.0000	41.0000	24.5000	29.0000	10.5000	5.0000
			Maximum	14.0000	17.5000	76.0000	77.0000	67.5000	44.5000	64.0000	68.0000

Table I-1. Average Mean Ranks (continued)

Scenario #				# of nav error-related replans generated	number of nav error-related replans accepted	number of threat-based replans generated	number of threat-based replans accepted	number of missiles locked on	number of missiles launched	percent of missiles defeated	range at release
Total Across All Scenarios	Operator	HITL	Mean	46.5313	44.0104	39.2917	38.1875	42.7708	46.5417	39.7354	46.6375
			Std Deviation	11.2301	11.6104	6.3619	4.4416	6.6121	5.5431	11.8491	14.5118
			Minimum	28.5000	22.5833	33.4167	35.2500	32.8333	38.1667	28.2500	26.5000
			Maximum	61.0000	56.5833	51.3333	47.0000	52.6667	53.2500	65.3333	68.5000
		HPM	Mean	37.1250	40.4861	46.7778	48.2500	42.1389	37.1111	42.3889	36.0833
			Std Deviation	7.9915	7.8002	4.6421	4.5166	2.3343	6.2837	6.4466	8.7957
			Minimum	23.8333	27.0833	39.5833	41.2500	38.9167	28.4167	31.1000	23.6667
			Maximum	44.5833	47.0000	51.9167	53.2500	45.7500	46.6667	48.4000	47.8333

APPENDIX J

CORRELATIONS AMONG THE DEPENDENT MEASURES

This page left blank intentionally

Table J-1. Correlations Among the Dependent Measures for the Entire Sample

Dependent Measures	DV1	DV2	DV3	DV4	DV5	DV6	DV7	DV8
DV1	XXXX							
DV2	0.976**	XXXX						
DV3	-0.140	0.114	XXXX					
DV4	-0.077	-0.060	0.891**	XXXX				
DV5	0.454**	0.470**	0.143	0.218*	XXXX			
DV6	0.489**	0.510**	0.203	0.225*	0.735**	XXXX		
DV7	-0.082	-0.073	-0.238*	-0.283*	0.064	-0.036	XXXX	
DV8	-0.163	-0.161	0.177	-0.271*	-0.236*	-0.240*	-0.050	XXXX

**p<.01
*p<.05

Table J-2. Correlations Among the Dependent Measures by Operator Type

Dependent Measures	DV1	DV2	DV3	DV4	DV5	DV6	DV7
	HPM HITL	HPM HITL	HPM HITL	HPM HITL	HPM HITL	HPM HITL	HPM HITL
DV2	1.0** .954**						
DV3	-.039 -.288*	-.039 -.266					
DV4	-.039 -.104	-.039 -.122	1.0** .638**				
DV5	.568** .332*	.568** .361*	.172 .116	.172 .320*			
DV6	.819** .327*	.819** .398**	.225 .323*	.225 .416**	.690** .826**		
DV7	-.211 .064	-.211 .075	-.590** .142	-.590** .009	-.006 .126	-.222 .077	
DV8	-.300 -.117	-.300 -.073	-.223 -.108	-.223 -.291*	-.442** -.127	-.50** -.224	.216 -.177

**p<.01

*p<.05